



Titre: Modèles linéaires pour la représentation et l'évaluation des
Title: compétences

Auteur: Rhouma Naceur
Author:

Date: 2013

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Naceur, R. (2013). Modèles linéaires pour la représentation et l'évaluation des
Citation: compétences [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/1133/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1133/>
PolyPublie URL:

**Directeurs de
recherche:** Michel Desmarais
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

MODÈLES LINÉAIRES POUR LA REPRÉSENTATION ET L'ÉVALUATION DES
COMPÉTENCES

RHOUMA NACEUR
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AVRIL 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MODÈLES LINÉAIRES POUR LA REPRÉSENTATION ET L'ÉVALUATION DES
COMPÉTENCES

présenté par : NACEUR Rhouma

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GALINIER Philippe, Doct., président

M. DESMARAIS Michel C., Ph.D., membre et directeur de recherche

Mme CARON France, Ph.D., membre

Je dédie mon mémoire à :

*Mon père Abde Ennaceur qui m'a tout donné, tout sacrifié, qu'il trouve dans ce mémoire le
fruit de ses effort.*

Ma mère Dalila, qui est toute ma vie, qui m'a soutenu dans mes choix et à qui je dois tous.

Mon frère et mes soeurs à qui je souhaite tout le bonheur du monde.

*Mon oncle Monsef pour tout ce qu'il m'a fait et toute l'aide qu'il m'a donné et surtout par
ce qu'il croit en moi.*

Tous mes amis

Mes professeurs dont revient le mérite de ma bonne formation

Tous ceux que j'aime

RHOUMA

REMERCIEMENTS

Je tiens à remercier en premier lieu mon directeur de recherche M. Michel C. Desmarais pour les conseils, les encouragements et toute l'aide qui m'a apporté durant mon projet de recherche.

Je tiens à remercier tout les membres de mon laboratoire pour leur soutien et leur conseils.

Enfin un grand merci à toutes les personnes qui m'ont aidé de près ou de loin dans la mise en oeuvre de ce projet.

Rhouma

RÉSUMÉ

Les environnements d'apprentissage reposent sur la modélisation et l'évaluation des compétences des étudiants. Une évaluation équitable et rapide de la connaissance de l'utilisateur de ces environnements permet d'adapter son contenu didactique et de le guider à travers la matière pertinente en indiquant celle qui est déjà maîtrisée et celle dont les pré-requis ne sont pas atteints. Cette évaluation est basée sur la prédiction de la performance des étudiants à partir de leurs réponses à un ensemble des questions. Un diagnostic des compétences de l'étudiant précis vise à le situer par rapport aux différents sujets et aux attentes et de lui donner les recommandations nécessaires pour qu'il puisse se rattraper. Afin de réaliser ce diagnostic, il est nécessaire d'établir l'alignement entre les questions et les compétences requises afin d'y répondre correctement.

En général la modélisation des compétences des étudiants est faite en se basant sur des modèles non linéaires comme les réseaux bayésiens ou sur le modèle de théorie de réponse aux items (IRT).

Dans notre recherche, nous visons étudier les modèles linéaires et pour ce faire nous avons choisi le modèle conjonctif et le modèle compensatoire, nous les avons comparés avec celui de IRT. La comparaison est faite avec le modèle IRT et basée sur la validité de prédiction. Nous avons utilisé également différents jeux de données.

Les simulations montrent que le modèle IRT donne toujours les meilleurs résultats, par contre nous avons constaté que les algorithmes hybrides que nous avons proposés (Algorithmes 5 et 6) ont donné de bons résultats. D'autre part, l'algorithme 5, qui est une combinaison entre les données générées avec IRT et les données d'entraînement, a donné sous certaines conditions des résultats meilleurs ou semblables à ceux de IRT. Les autres algorithmes ont donné des résultats acceptables avec les deux modèles conjonctif et compensatoire. Le modèle conjonctif a donné des résultats meilleurs que ceux du modèle compensatoire sous certaines conditions.

ABSTRACT

Learning environments are based on the modeling and assessment of student's skills. This assessment rely on the prediction of the performance of students based on their responses to a set of questions. A diagnostic skills of the student opts to give the results of the knowledge of skills based on their responses, and recommendations improve his skills. To make this diagnostic, it is necessary to establish the alignment between questions and skills in order to respond appropriately.

Generally modeling skills of students is done using non-linear models such as Bayesian network. However, regarding the linear models, most of the research based on the model response theory questions (IRT). In our research, we aimed to study the linear models and for that reason we chosed the conjunctive model and the compensatory model, we also compared them with the IRT. The comparison is based on the predictive validity, hence we have proposed some algorithms in order to investigate the predictive validity of our models. We also used different data sets for more precision in terms of analysis and conclusion.

The simulations showed that the IRT model remains the one that gives the best results. we found that against the proposed hybrid algorithms (Algorithms 5 and 6) give good results. We found that the proposed hybrid algorithm, which is a combination of data generated with IRT and training data has been given under certain conditions better results or almost equal to those of IRT. The other algorithms have given acceptable results with both models conjunctive and compensatory. In the other side and despite its complexity, the conjunctive model gave significantly better results than compensatory model under certain conditions.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xiv
LISTE DES ANNEXES	xvi
LISTE DES SIGLES ET ABRÉVIATIONS	xvii
CHAPITRE 1 INTRODUCTION	1
1.1 Cadre de la recherche	1
1.2 Objectifs de recherche	2
1.3 Organisation du mémoire	2
CHAPITRE 2 REVUE DE LITTÉRATURE	3
2.1 Exemple d'évaluation de compétences	3
2.2 Compromis entre les modèles d'apprentissage	4
2.3 Les environnements d'apprentissage	4
2.3.1 Tutoriels cognitifs et Modèles à Base des Contraintes	4
2.3.2 Séquencement du contenu adaptatif	6
2.4 Les modèles probabilistes de compétences	7
2.4.1 Réseaux bayésiens et modèles graphiques	8
2.4.2 Théorie de réponses aux items	8
2.4.3 Structure de connaissance avec ordre partiel : POKS	10
2.5 Retour sur les modèles de compétences	11

CHAPITRE 3	MODÈLES MATRICIELS DE COMPÉTENCE	13
3.1	Alignement Item/Compétence, Étudiant/Compétence et Étudiant/item . . .	13
3.1.1	Alignement Étudiant/Compétence	13
3.1.2	Alignement Étudiant/item	14
3.1.3	Alignement Item/Compétence : matrice Q	14
3.2	Factorisation des matrices en valeurs non négatives	15
3.3	Modèles d'alignement des questions et des compétences	16
3.3.1	Le modèle compensatoire	16
3.3.2	Modèle conjonctif	17
3.4	Factorisation et régression linéaire	17
3.5	Algorithmes proposés	19
3.5.1	Algorithme 1	19
3.5.2	Algorithme 2	21
3.5.3	Algorithme 3	22
3.5.4	Algorithme 4	22
3.5.5	IRT et modèle des valeurs attendues	23
3.5.6	Algorithme 5	24
3.5.7	Algorithme 6	25
CHAPITRE 4	RÉSULTATS ET ANALYSES	27
4.1	Présentation des données	27
4.2	Déroulement des simulations	28
4.3	Résultats du modèle compensatoire	28
4.3.1	Résultats obtenus avec les données de Vomlel	28
4.3.2	Résultats obtenus avec les données d'Unix	29
4.3.3	Résultats obtenus avec les données de Polytechnique	32
4.4	Résultats avec le modèle conjonctif	32
4.4.1	Résultats obtenus avec les données de Vomlel	32
4.4.2	Résultats obtenus avec les données d'Unix	34
4.4.3	Résultats obtenus avec les données de Polytechnique	36
4.5	Analyse des résultats	38
4.5.1	Analyses pour les résultats avec le modèle compensatoire	38
4.5.2	Analyses pour les résultats avec le modèle conjonctif	40
4.5.3	Analyses de comparaison de modèle compensatoire et modèle conjonctif	42
4.5.4	Analyses générales	45

CHAPITRE 5 CONCLUSION	48
5.1 Synthèse du déroulement du projet	48
5.2 Perspectives et travaux futurs	49
RÉFÉRENCES	50
ANNEXES	52

LISTE DES TABLEAUX

Tableau 4.1	Résumé des jeux de données	27
Tableau 4.2	Tableau des résultats de performance avec les données de Vomlel du modèle compensatoire, n.skills = 5 et obs.size = 5.	29
Tableau 4.3	Tableau des résultats de performance avec les données de Vomlel du modèle compensatoire, n.skills = 5 et obs.size = 10	30
Tableau 4.4	Tableau des résultats de performance avec les données de Vomlel du modèle compensatoire, n.skills = 5 et obs.size = 15	30
Tableau 4.5	Tableau des résultats de performance avec les données de Unix du modèle compensatoire, n.skills = 5 et obs.size = 10	30
Tableau 4.6	Tableau des résultats de performance avec les données de Unix du modèle compensatoire, n.skills = 5 et obs.size = 20	31
Tableau 4.7	Tableau des résultats de performance avec les données de Unix du modèle compensatoire, n.skills = 5 et obs.size = 25	31
Tableau 4.8	Tableau des résultats de performance avec les données de Polytechnique du modèle compensatoire, n.skills = 5 et obs.size = 15	33
Tableau 4.9	Tableau des résultats de performance avec les données de Polytechnique du modèle compensatoire, n.skills = 5 et obs.size = 30	33
Tableau 4.10	Tableau des résultats de performance avec les données de Polytechnique du modèle compensatoire, n.skills = 5 et obs.size = 45	33
Tableau 4.11	Tableau des résultats de performance avec les données de Vomlel du modèle conjonctif, n.skills = 5 et obs.size = 5	34
Tableau 4.12	Tableau des résultats de performance avec les données de Vomlel du modèle conjonctif, n.skills = 5 et obs.size = 10	35
Tableau 4.13	Tableau des résultats de performance avec les données de Vomlel du modèle conjonctif, n.skills = 5 et obs.size = 15	35
Tableau 4.14	tableau des résultats de performance avec les données de Unix du modèle conjonctif, n.skills = 5 et obs.size = 10	35
Tableau 4.15	Tableau des résultats de performance avec les données de Unix du modèle conjonctif, n.skills = 5 et obs.size = 20	36
Tableau 4.16	tableau des résultats de performance avec les données de Unix du modèle conjonctif, n.skills = 5 et obs.size = 25	37
Tableau 4.17	Tableau des résultats de performance avec les données de Polytechnique du modèle conjonctif, n.skills = 5 et obs.size = 15	37

Tableau 4.18	tableau des résultats de performance avec les données de Polytechnique du modèle conjonctif, n.skills = 5 et obs.size = 30	37
Tableau 4.19	Tableau des résultats de performance avec les données de Polytechnique du modèle conjonctif, n.skills = 5 et obs.size = 45	38
Tableau 4.20	Tableau comparatif des résultats de taux d'erreurs pour les données de Vomlel (modèle compensatoire)	39
Tableau 4.21	Tableau comparatif des résultats de taux d'erreurs pour les données de Unix (modèle compensatoire)	39
Tableau 4.22	Tableau comparatif des résultats de taux d'erreurs pour les données de Polytechnique (modèle compensatoire)	40
Tableau 4.23	Tableau comparatif des résultats de taux d'erreurs pour les données de Vomlel (modèle conjonctif)	40
Tableau 4.24	Tableau comparatif des résultats de taux d'erreurs pour les données d'Unix (modèle conjonctif)	41
Tableau 4.25	Tableau comparatif des résultats de taux d'erreurs pour les données de Polytechnique (modèle conjonctif)	42
Tableau 4.26	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Vomlel avec ops.size=5	42
Tableau 4.27	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Vomlel avec ops.size=10	43
Tableau 4.28	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Vomlel avec ops.size=15	43
Tableau 4.29	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données d'Unix avec ops.size=10	44
Tableau 4.30	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données d'Unix avec ops.size=20	44
Tableau 4.31	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données d'Unix avec ops.size=25	45

Tableau 4.32	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Polytechnique avec ops.size=15	46
Tableau 4.33	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Polytechnique avec ops.size=30	46
Tableau 4.34	Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Polytechnique avec ops.size=45	46
Tableau C.1	Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.1	57
Tableau C.2	Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.2	59
Tableau C.3	Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.3	60
Tableau C.4	Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.4	61
Tableau C.5	Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.5	62
Tableau C.6	Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.6	63
Tableau C.7	Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.7	64
Tableau C.8	Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.8	66
Tableau C.9	Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.9	67
Tableau C.10	Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.10	68
Tableau C.11	Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.11	69
Tableau C.12	Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.12	70
Tableau C.13	Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.13	71

Tableau C.14	Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.14	72
Tableau C.15	Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.15	73
Tableau C.16	Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.16	74
Tableau C.17	Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.17	75
Tableau C.18	Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.18	76

LISTE DES FIGURES

Figure 2.1	Exemple de Connaissances des règles de traçage et des contraintes de CBM	5
Figure 2.2	Exemple de réseau bayésien BN	9
Figure 2.3	Exemple d'un IRT avec une compétence unique	10
Figure 2.4	Exemple de graphe de POKS	11
Figure 2.5	Graphe de POKS avec le DAG	11
Figure 3.1	Exemple de matrices d'alignements.	14
Figure 3.2	Exemple des matrices V, W et H.	15
Figure 3.3	Exemple d'une matrice Q	17
Figure 3.4	Décomposition des données	18
Figure 3.5	Explication du principe de séparation des données	20
Figure 3.6	Principe de génération de la matrice des données de l'algorithme 5 . .	25
Figure 3.7	Principe de génération de la matrice des données de l'algorithme 6 . .	26
Figure A.1	Résultats du test fait par l'étudiant	53
Figure A.2	Courriel transmis aux étudiants après le test	54
Figure C.1	Résultats de performance avec les données de Vomlel avec le modèle compensatoire, n.skills = 3 et obs.size = 5	58
Figure C.2	Résultats de performance avec les données de Vomlel avec le modèle compensatoire, n.skills = 3 et obs.size = 10	58
Figure C.3	Résultats de performance avec les données de Vomlel avec le modèle compensatoire, n.skills = 3 et obs.size = 15	59
Figure C.4	Résultats de performance avec les données de Unix avec le modèle compensatoire, n.skills = 3 et obs.size = 10	60
Figure C.5	Résultats de performance avec les données de Unix avec le modèle compensatoire, n.skills = 3 et obs.size = 20	61
Figure C.6	Résultats de performance avec les données de Unix avec le modèle compensatoire, n.skills = 3 et obs.size = 25	62
Figure C.7	Résultats de performance avec les données de Polytechnique avec le modèle compensatoire, n.skills = 3 et obs.size = 15	63
Figure C.8	Résultats de performance avec les données de Polytechnique avec le modèle compensatoire, n.skills = 3 et obs.size = 30	64
Figure C.9	Résultats de performance avec les données de Polytechnique avec le modèle compensatoire, n.skills = 3 et obs.size = 45	66

Figure C.10	Résultats de performance avec les données de Vomlel avec le modèle conjonctif, n.skills = 3 et obs.size = 5	67
Figure C.11	Résultats de performance avec les données de Vomlel avec le modèle conjonctif, n.skills = 3 et obs.size = 10	68
Figure C.12	Résultats de performance avec les données de Vomlel avec le modèle conjonctif, n.skills = 3 et obs.size = 15	69
Figure C.13	Résultats de performance avec les données de Unix avec le modèle conjonctif, n.skills = 3 et obs.size = 10	70
Figure C.14	Résultats de performance avec les données de Unix avec le modèle conjonctif, n.skills = 3 et obs.size = 20	71
Figure C.15	Résultats de performance avec les données de Unix avec le modèle conjonctif, n.skills = 3 et obs.size = 25	72
Figure C.16	Résultats de performance avec les données de Polytechnique avec le modèle conjonctif, n.skills = 3 et obs.size = 15	73
Figure C.17	Résultats de performance avec les données de Polytechnique avec le modèle conjonctif, n.skills = 3 et obs.size = 30	74
Figure C.18	Résultats de performance avec les données de Polytechnique avec le modèle conjonctif, n.skills = 3 et obs.size = 45	75

LISTE DES ANNEXES

Annexe A	Exemple du résultat d'un test	52
Annexe B	Exemple du questionnaire de polytechnique	55
Annexe C	D'autres résultats	57
Annexe D	Code R des algorithmes	77

LISTE DES SIGLES ET ABRÉVIATIONS

CBM	Constraint Based Modeling
KT	Knowledge Tracing rules
CT	Cognitive Tutors
BN	Bayesian Network
IRT	Item Response Theory
POKS	Partial Order Knowledge Structures
DAG	Directed Acyclic Graph
NMF	Non-negative matrix factorization

CHAPITRE 1

INTRODUCTION

1.1 Cadre de la recherche

Ce mémoire porte sur le développement et la validation d'algorithmes qui visent à découvrir, à partir de données empiriques, les compétences impliquées pour réaliser différentes épreuves cognitives. Il s'inscrit dans le cadre d'un projet de développement d'un guide d'étude parrainé par la Maison des technologies de formation et d'apprentissage (MATI). La MATI a pour mission de développer des connaissances, des méthodes et des outils pour l'utilisation des Technologies d'Information et de Communication dans l'enseignement et l'apprentissage des sciences, des mathématiques, du génie, de la gestion et des disciplines connexes.¹

Le guide d'étude est une application qui présente un ensemble de questions et d'exercices à des étudiants, et les résultats permettent au système d'évaluer des compétences sur la base des inférences [Desmarais *et al.* (2012)]. Au long de l'utilisation du guide d'étude, l'étudiant obtient un retour d'information quant à son progrès et à celui qu'il lui reste à faire en vue d'être adéquatement préparé à un cours, un examen, ou toute autre activité nécessitant une préparation.

Par conséquent, cette application permet non seulement de mieux mesurer le niveau de préparation acquis par l'étudiant, mais en plus, elle lui permet de concentrer son temps d'étude aux sujets les plus pertinents. Pour y arriver, un tel service nécessite un diagnostic des compétences de l'étudiant afin de le situer par rapport aux différents sujets et aux attentes. Ce diagnostic doit s'effectuer le plus précisément et rapidement possible, en minimisant le nombre de questions. Lorsqu'une question touche à plusieurs sujets, une façon efficace de minimiser les questions consiste à utiliser le résultat pour chacun de ces sujets. Il faut donc déterminer quels sujets, ou compétences, se retrouvent impliqués derrière chaque question.

Le processus d'aligner chaque question par rapport à des compétences/sujets donnés peut se concevoir comme celui de définir une matrice de m lignes représentant les questions et de n colonnes représentant les compétences, où une valeur non nulle indique qu'une compétence est impliquée dans la réponse à la question. Cette matrice s'appelle la matrice Q .

L'objectif de notre mémoire est d'aider à définir cette matrice Q à partir de données constituées de résultats d'un ensemble d'individus ayant répondu aux questions. Différentes techniques de décomposition de matrices seront explorées à cette fin.

1. www.matimtl.ca

1.2 Objectifs de recherche

L'objectif principal de ce mémoire est de développer et de valider différents algorithmes pour établir quelles sont les compétences impliquées dans la réalisation de tâches cognitives, plus particulièrement la réponse à des questions mathématiques.

D'une manière plus spécifique nous visons les objectifs suivants :

- Déterminer les méthodes efficaces pour établir l'alignement des questions et des compétences.
- Proposer quelques algorithmes et évaluer leurs validités de prédiction.
- Comparer les résultats et déterminer le modèle le plus efficace.

À noter que nous ne faisons pas la distinction entre des compétences et des connaissances dans le cadre de ce travail et que les deux termes sont parfois utilisés de façon interchangeable. De manière générale, les compétences se rapportent souvent à des aptitudes procédurales tandis que les connaissances se rapportent plutôt à des informations déclaratives que l'étudiant possède. Mais dans un cas comme dans l'autre, ce sont des acquis nécessaires à répondre correctement à des questions ou à des exercices.

1.3 Organisation du mémoire

Ce mémoire est divisé en quatre chapitres.

Le premier chapitre porte sur la revue des théories et modèles pertinents. Nous présenterons dans ce chapitre les méthodes les plus importantes utilisées dans les systèmes de recommandation.

Le deuxième chapitre porte sur les modèles matriciels de compétence. Dans ce chapitre nous présentons les détails des algorithmes que nous proposons pour chaque modèle.

Le troisième chapitre porte sur les résultats et analyses. Ce chapitre présente le fruit de notre recherche, nous expliquons également les résultats trouvés et détaillons les analyses que nous avons faites.

Nous terminons ce rapport par une conclusion dans laquelle nous ferons une récapitulation de tout ce que nous avons fait au long du projet et les défis que nous avons trouvés. Nous présentons dans une deuxième partie quelques perspectives ainsi que nos travaux futurs.

CHAPITRE 2

REVUE DE LITTÉRATURE

Les environnements d'apprentissage sont de plus en plus répandus dans nos systèmes d'éducation modernes. L'évaluation des connaissances des individus constitue un élément critique au succès de ces environnements. Idéalement, ces environnements devraient déterminer précisément le niveau de connaissance d'un individu à partir de tests ou de l'observation du comportement de l'étudiant. L'évaluation des connaissances devrait être le plus parcimonieuse et efficace possible : un trop grand nombre de questions pour en arriver à une évaluation précise peut démotiver l'étudiant, tandis qu'une évaluation trop approximative ou imprécise peut compromettre la démarche d'apprentissage. Ces systèmes doivent donc reposer sur un modèle de connaissance qui leur permet d'effectuer ce diagnostic précis et efficace des compétences acquises. Les tests adaptatifs assistés par ordinateur sont un exemple de système qui comporte un modèle de connaissance. L'avantage d'un système de test assisté par ordinateur est de pouvoir minimiser la longueur des questionnaires et le nombre des questions du test en déterminant quelles compétences sont requises pour chaque question.

Dans ce chapitre, nous présentons les différents modèles et techniques utilisés dans les environnements d'apprentissage pour modéliser et évaluer la connaissance. Nous expliquons dans une deuxième partie le principe d'alignement des questions avec les compétences requises.

2.1 Exemple d'évaluation de compétences

Dans le but de comprendre le principe de l'évaluation des compétences et connaissances acquises, prenons un exemple précis. Si la réponse de l'étudiant à une question pour trouver la somme de $(\frac{1}{3} + \frac{3}{4})$ est $\frac{4}{7}$, l'expert, ou l'enseignant peut facilement analyser que l'étudiant a fait la somme des deux numérateurs ensemble et des deux dénominateurs ensemble ($\frac{1+3}{3+4}$). Ce qui lui permet de se focaliser avec l'étudiant sur les deux étapes à suivre pour résoudre le problème, à savoir déterminer le plus petit dénominateur commun et l'addition du numérateur de fractions qui ont le même dénominateur pour en arriver à la réponse ($\frac{1 \times 4 + 3 \times 3}{3 \times 4}$). On peut donc considérer la notion de dénominateur commun et la règle qui permet d'additionner les numérateurs de fractions qui ont le même dénominateur comme les deux compétences requises pour résoudre le problème. Un élève qui ne possède pas ces compétences obtiendra une mauvaise réponse à ce type de problèmes. D'autres facteurs peuvent aussi entrer en jeu,

comme l'état psychologique de l'étudiant, son état de fatigue, s'il est stressé.

2.2 Compromis entre les modèles d'apprentissage

Les environnements d'apprentissage et les techniques de modélisation diffèrent entre eux selon plusieurs critères et dimensions. Ces dimensions sont les suivantes[Desmarais *et al.* (2006)] :

- Flexibilité et expressivité
- Coût de la définition du modèle
- Évolutivité
- Coût de la mise à jour
- Exactitude et la fiabilité de la prédiction
- Fiabilité et sensibilité à des facteurs externes
- Base mathématique
- Approximations, postulats et hypothèses.

Tous ces facteurs peuvent être considérés comme des métriques permettant d'évaluer l'utilité d'une approche ou technique et de comparer entre eux. Idéalement, nous cherchons un modèle d'apprentissage qui suffit à tous ces facteurs. Une telle approche limitera l'effort de construction de modèle à celle de la collecte des données. Elle facilitera la mise à jour, comme dans le cas d'ajout d'un nouveau concept, cela nécessitera seulement de relancer l'algorithme d'apprentissage qu'on peut le faire en temps réel lors du rassemblement des données. Finalement, si la construction du modèle est basée sur une approche algorithmique, la fiabilité et la précision, seront moins dépendants des facteurs comme l'expertise et la subjectivité de celui qui définit le modèle.

2.3 Les environnements d'apprentissage

Les systèmes de Tutoriels Intelligents sont très utilisés dans l'éducation. Ils connaissent un succès important. Dans cette section nous présentons les familles des systèmes de tutoriels intelligents les plus connus et leurs modèles de représentation des compétences.

2.3.1 Tutoriels cognitifs et Modèles à Base des Contraintes

Les Tutoriels Cognitifs (Cognitive Tutors CT)[Koedinger *et al.* (1997); Corbett et Anderson (1995)] représentent la compétence procédurale qui peut être directement associée aux actions de l'étudiant¹. Les Modèles à Base des Contraintes (Constraint Based Modeling

1. Ces modèles sont commercialisés par Carnegie Learning Inc

CBM)[Mitrovic (2012)], quant à eux, représentent plutôt la compétence déclarative comme un ensemble de contraintes selon la réponse de l'étudiant ou selon les résultats de l'action de l'étudiant.

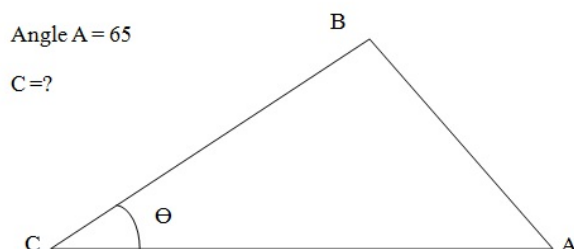


Figure 2.1 Exemple de Connaissances des règles de traçage et des contraintes de CBM

La figure 2.1[Mitrovic *et al.* (2003)] illustre quelques exemples des Connaissances des règles de traçage (Knowledge Tracing rules KT) et des prédicats pour l'approche CBM pour un problème relatif au calcul des angles d'un triangle isocèle[Desmarais et Baker (2011)].

Connaissances des règles de traçage (Knowledge Tracing rules KT)

Les règles de traçage s'expriment sous la forme de règles de production :

1. Si le but est de déterminer un angle dans un triangle isocèle ABC avec $AC = BC$ et l'angle A est connu, alors attribuer la valeur de l'angle A à B
2. Si le but est de déterminer un angle dans un triangle isocèle ABC et les angles A et B sont connus, alors attribuer à l'angle C la valeur de $(180-A-B)$
3. Si le but est de déterminer un angle dans un triangle isocèle ABC et que les angles A et C sont en bas du triangle et que l'angle A est connu, alors associer la valeur de l'angle A à C

Les règles de production comportent des prémisses qui, si elles s'avèrent vrai, génèrent des conséquences.

De ces trois règles, les deux premières sont correctes. Par contre, la troisième est en fait une **fausse conception**. Il est donc ainsi possible de représenter dans le modèle KT des connaissances procédurales valables sous la forme prémisses→conséquences, et de représenter des fausses conception qui se reflèteront à travers des erreurs.

Les contraintes CBM

Dans le cas des contraintes CBM, les connaissances sont plutôt représentées sous une forme déclarative en mettant l'accent sur les relations entre les éléments du problème et les

étapes pour déterminer l'information à obtenir :

- C_{r1} : un angle de base d'un triangle isocèle est connu (θ_1) et l'étudiant a déterminé la valeur de l'autre angle de base (θ_2)
- C_{s1} : l'angle $\theta_1 = \theta_2$
- C_{r2} : un angle de base dans un triangle isocèle θ_1 est connu et l'étudiant a déterminé la valeur de l'autre angle de base θ_2 qui est égale à θ_1
- C_{s2} : θ_2 est un angle de base
- C_{r3} : deux angles du triangle sont connus (θ_1 et θ_2) et l'étudiant a calculé la valeur du troisième angle θ_3
- C_{s3} : la valeur de l'angle θ_3 est égale à $(180 - \theta_1 - \theta_2)$

Avec ce modèle, les fausses conceptions sont représentées par une violation de contraintes. Par exemple, dans la figure 2.1, l'application de la troisième règle entraînerait une violation de la contrainte C_{s3} .

Malgré leurs différences, les deux approches CT et CBM ont beaucoup de similarités et génèrent des résultats presque identiques [Desmarais et Baker (2011)]. En effet, avec CT la compétence est présentée comme une règle, par contre avec le CBM elle est présentée comme un prédicat. Ces deux définitions de la compétence pour les deux approches sont fortement similaires.

Un autre point de similarité, c'est qu'avec le CT, la compétence est maîtrisée si la règle est liée à la performance de l'action de l'étudiant. Avec le CBM, une compétence est maîtrisée si le prédicat est lié aux réponses de l'étudiant.

Pour les deux approches CT et CBM, quand une mauvaise connaissance est détectée, cette information permet au tuteur de donner à l'étudiant les directives immédiates et nécessaires afin de se rattraper. Cette réaction de rattrapage juste-à-temps est un facteur crucial du succès du processus d'apprentissage de l'étudiant.

Cette analyse est basée sur l'action de l'utilisateur pour tracer la résolution du problème s'appelle le Modèle de Traçage. Elle permet au tuteur de donner des commentaires et des recommandations et est similaire au processus pour identifier quelle contrainte a été satisfaite dans l'approche CBM [Desmarais et Baker (2011)].

2.3.2 Séquencement du contenu adaptatif

Parmi les tutoriels qui connaissent un grand succès de nos jours, nous trouvons les environnements qui servent à guider l'étudiant à travers le matériel didactique (content sequencing tutorials). Parmi les environnements les plus répandus, nous trouvons ALEKS² qui est utilisé par plusieurs centaines de milliers d'utilisateurs.

2. www.aleks.com

ALEKS est l'aboutissement des travaux de [Doignon et Falmagne (1985)], entre autres, qui ont pris la forme d'un guide d'étude commercialisé par l'Université de Californie [Falmagne *et al.* (2006)]. Il est basé sur la théorie cognitive des espaces de connaissance. Il combine plusieurs approches tutoriels. Ces approches peuvent être considérées comme une séquence curriculum (Curriculum Sequencing), qui est un concept qui consiste à définir un chemin d'apprentissage dans un espace d'objectifs d'apprentissage et de contenu didactique [Desmarais et Baker (2011)]. Plusieurs systèmes utilisent l'approche de séquençement de contenu, comme SIETTE [Conejo *et al.* (2004)] et RATH [Hockemeyer *et al.* (1997)].

La différence entre cette approche et les deux autres approches que nous avons présentées précédemment, le CT et le CBM, c'est que le CT et le CBM fournissent les recommandations nécessaires basées sur le détail des analyses qui portent sur les réponses des étudiants pour un problème spécifique. Alors que l'approche basée sur le séquençement de contenu vise à adapter le contenu didactique de manière globale.

L'atout majeur des tutoriels visant l'analyse de la solution comme les tutoriels qui utilisent l'approche CT et CBM est l'aspect "juste-à-temps" des recommandations qui permettent aussi de connaître le bon moment pour décider d'avancer dans le processus d'apprentissage et de passer à un autre thème.

2.4 Les modèles probabilistes de compétences

Plusieurs facteurs doivent être considérés lors de la modélisation et l'assignation des compétences. Cependant, il est crucial de déterminer après combien de cas de succès nous pouvons juger qu'une compétence est bien maîtrisée. La réponse à cette question est difficile, elle dépend éventuellement de plusieurs facteurs. L'occurrence de *slip* (quand un étudiant donne une mauvaise réponse malgré qu'il connaisse le sujet) et le *guess* (quand un étudiant répond correctement à une question malgré qu'il n'ait pas les compétences requises) sont les facteurs les plus importants de l'incertitude [Desmarais et Baker (2011)].

Dans cette section nous révisons les approches probabilistes pour la modélisation des compétences qui prennent en considération l'incertitude comme facteur important à inclure lors de la prédiction.

L'incertitude est particulièrement importante lors d'une évaluation globale où la preuve est indirecte. Par exemple, si un étudiant trouve la réponse à $\sqrt{(2^6)} = 8$, un modèle de transfert peut conclure que l'étudiant sera capable de résoudre le problème $(\frac{3}{4} \times \frac{7}{5})$. Par contre, comme les compétences requises sont différentes pour chaque problème, nous pouvons affirmer que cette évidence est indirecte et incertaine.

2.4.1 Réseaux bayésiens et modèles graphiques

Le réseau bayésien (Bayesian Network BN) est un modèle graphique probabiliste très flexible. Les BNs sont largement utilisés dans la modélisation des compétences des apprenants. Les BNs sont utilisés par différents environnements d'apprentissage intelligents comme ANDE[Conati *et al.* (2002); VanLehn *et al.* (2005)], HYDRIVE[Mislevy et Gitomer (1995)]. Le point fort des réseaux bayésiens réside dans leur capacité de représentation élevée et le fait qu'ils se prêtent à une représentation graphique intuitive. De plus, les BNs offrent un formalisme bien défini qui se prête à effectuer des calculs de probabilité des noeuds non observés à partir des informations sur les noeuds observés. Plusieurs applications et bibliothèques existent, permettant le calcul de la probabilité basée sur les noeuds observés. Par contre, dans la pratique, l'application des BNs est complexe, car, il est nécessaire de déterminer la topologie du réseau et les paramètres nécessaires pour calculer la probabilité des noeuds cachés.

La figure 2.2 illustre un réseau bayésien potentiel dans le domaine de l'arithmétique des fractions. Les compétences et les concepts impliqués (en haut) sont liés à des éléments spécifiques (en bas). Certains éléments nécessitent une seule compétence alors que d'autres impliquent deux compétences ou plus. De plus, les compétences, elles-mêmes, peuvent être liées entre elles, exprimant ainsi un ordre de compétences croissantes. Par exemple $C1 \rightarrow C2$ et $C1 \rightarrow C3$.

Les liens représentent les interdépendances qui sont présentées sous forme d'une probabilité conditionnelle entre un noeud enfant étant donné ses noeuds parents $P(fils|parents)$. En suivant la sémantique des BNs, on peut conclure à partir de la figure 2.2 que si l'étudiant connaît la réponse à C2, cela augmente la chance que C3 soit connu aussi. Par contre, si C1 est connu et après qu'on découvre que C2 est connu, cela n'augmentera pas nos convictions à propos de C3. Cette propriété est la Couverture de Markov, qui constitue le concept fondamental de BN. La topologie des BNs ne spécifie donc pas seulement les noeuds enfants et les noeuds parents, mais aussi l'indépendance conditionnelle entre les noeuds par le principe de la Couverture de Markov.

2.4.2 Théorie de réponses aux items

La théorie de Réponses aux items (Item Response Theory IRT) est une autre approche d'évaluation des connaissances issue du domaine de la psychométrie il y a plus de 40 ans. Elle constitue probablement le premier exemple d'environnement informatique pour l'évaluation personnalisée[Birnbaum (1968)]. En se basant sur la figure 2.3, IRT peut être représentée comme un seul noeud caché qui est lié à tous les autres noeuds observés [Desmarais et Baker (2011)]. Le principe de l'IRT c'est que la chance de réussir un exercice ou une question est

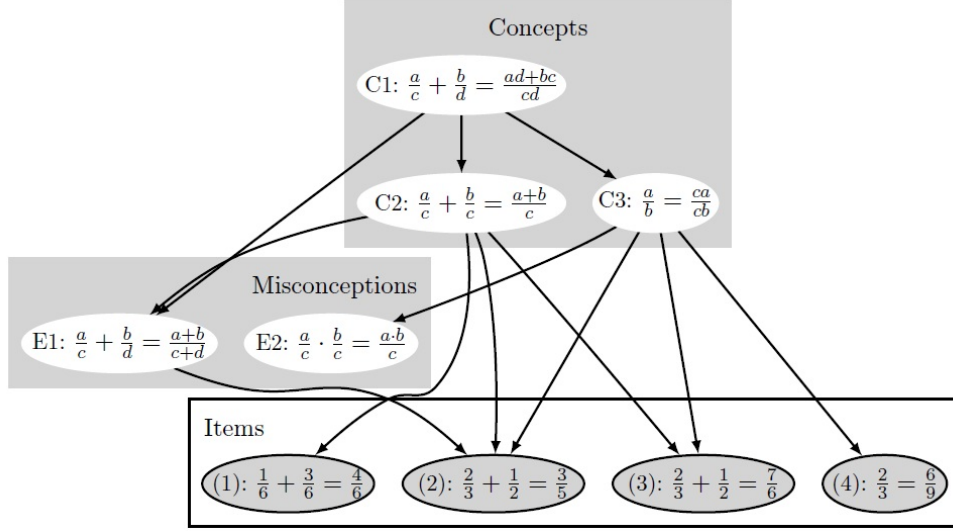


Figure 2.2 Exemple de réseau bayésien BN

proportionnelle au niveau de maîtrise du facteur latent, le noeud non-observable θ .

IRT peut être présenté comme un modèle de régression logistique : étant donné un vecteur des réponses X , la compétence maîtrisée θ est estimée en maximisant l'équation suivante :

$$P(\theta|X) = \prod_i P(X_i|\theta)$$

où $P(X_i|\theta)$ est représentée par une fonction logistique.

L'IRT est un modèle à compétence unique (Figure 2.3). Ce modèle est fondé sur une solide base théorique aussi bien en psychologie qu'en mathématiques [Desmarais et Baker (2011)]. D'autres modèles IRT existent afin de résoudre les problèmes avec deux compétences (noeuds cachés) ou plus. Ces modèles sont appelés IRT multi-dimensionnels. Ils ne sont pas fréquemment utilisés dans les environnements d'apprentissage personnalisés et sont caractérisés par leur plus grande complexité.

En IRT la probabilité du succès d'un seul élément X_i est déterminée par la fonction logistique :

$$P(X|\theta) = \frac{1}{(1 + e^{-a_i(\theta - b_i)})}$$

où a_i est le paramètre de discrimination et b_i est le paramètre de difficulté de l'item. Ces deux paramètres sont spécifiques à chaque item.

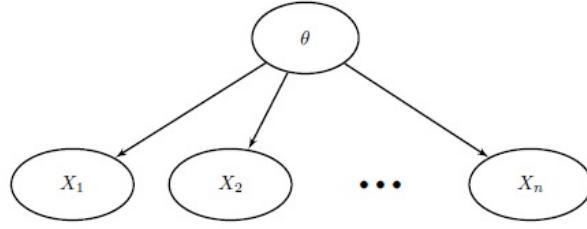


Figure 2.3 Exemple d'un IRT avec une compétence unique

2.4.3 Structure de connaissance avec ordre partiel : POKS

La structure de connaissance avec ordre partiel (Partial Order Knowledge Structures : POKS) est fondée sur le principe d'ordonnancement entre les questions. En effet, l'idée est que si un étudiant réussit à répondre à un ensemble des questions complexes, donc forcément il pourra répondre correctement à des questions moins difficiles, ou simples. D'un autre côté, si l'étudiant échoue une question simple, alors il sera forcément incapable de répondre à des questions plus difficiles [Desmarais *et al.* (2006)]. Les arcs d'un graphe POKS représentent en quelque sorte l'ordre d'apprentissage habituel [Desmarais *et al.* (2007)].

La figure 2.4 présente un exemple d'espace de connaissance avec l'ordre partiel. En effet, d'après la figure 2.4 on remarque que si l'étudiant répond à (c) alors il peut forcément répondre à (d). Puis, si l'étudiant échoue (c), il va probablement échouer (a). Par contre, l'élément (c) ne donne aucune information sur (b). Cette structure définit les états de connaissances possibles : $\{\emptyset, \{d\}, \{c, d\}, \{b, d\}, \{b, c, d\}, \{a, b, c, d\}\}$

Les sous-ensembles ci-dessus peuvent être présentés sous la forme d'un graphe acyclique orienté (Directed Acyclic Graph DAG)(figure 2.5).

La construction d'un graphe POKS s'effectue à partir de données empiriques[Desmarais et Gagnon (2006)]. Par exemple, si on cherche à déterminer si la relation $A \succ B$ est vraie, trois tests sont effectués :

$$P(B|A) \geq p_c$$

$$P(\bar{A}|\bar{B}) \geq p_c$$

$$P(B|A) \neq P(B)$$

On veut par le premier test vérifier si B est réussi sachant que A est réussi en effectuant un test d'hypothèse pour valider que $P(B|A)$ est supérieur à un seuil de probabilité. Pour le second test, on veut vérifier que si un échec pour A entraîne un échec pour B. Enfin, le troisième test vérifie que A a une influence sur B. Les trois conditions sont vérifiées en

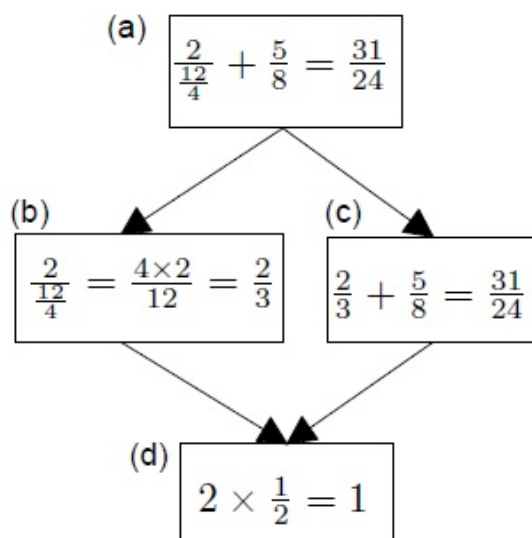


Figure 2.4 Exemple de graphe de POKS

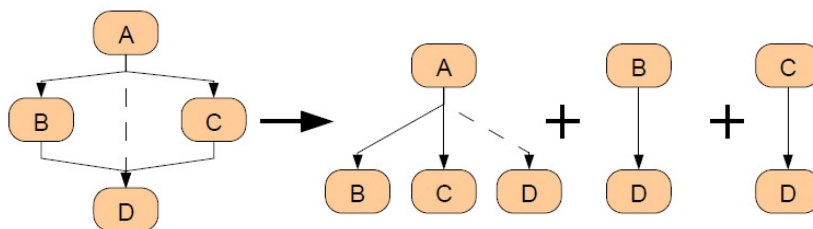


Figure 2.5 Graphe de POKS avec le DAG

utilisant un test d'hypothèse statistique en prenant $p_c = 0.5$ pour les deux premiers tests et en effectuant un test d'indépendance pour la troisième condition. Le réseau de POKS peut fournir de meilleures prédictions que celui de réseau bayésien [Desmarais *et al.* (2006, 2007)].

2.5 Retour sur les modèles de compétences

Il existe donc plusieurs modèles de compétences. Certains sont basés sur une représentation très détaillée sous la forme de règles ou de contraintes. D'autres sont basés sur une représentation probabiliste et peuvent aussi comporter une représentation détaillée de connaissances ou de fausses conceptions, notamment les réseaux bayésiens. Finalement certains sont probabilistes mais ne représentent qu'une seule compétence à la fois. C'est le cas du modèle IRT.

Plus les compétences sont détaillées, meilleur sera le diagnostic cognitif et la qualité de l'intervention d'un tuteur. Par contre, qu'un modèle détaillé soit à base de règles, de contraintes ou de noeuds dans un modèle bayésien, sa construction repose sur un travail expert d'ingénierie des compétences. Il est ainsi soumis à la qualité de l'expertise et aux efforts investis pour le travail d'ingénierie.

Au contraire, le modèle IRT est beaucoup moins détaillé avec sa seule et unique compétence, qui peut cependant prendre une valeur sous une échelle continue, mais il ne nécessite pas d'intervention experte. Il repose au contraire sur des données ce qui rend la modélisation d'un domaine de compétence plus facilement répliquable dans la mesure où l'on a les données de tests.

Le défi est donc de trouver un modèle qui permet à la fois une modélisation détaillée des compétences mais peut être construit à partir de données, sans nécessiter le recours à un expert. Le chapitre suivant aborde les modèles linéaires de compétences qui sont un pas vers la construction de modèles à plusieurs compétences construit à partir de données.

CHAPITRE 3

MODÈLES MATRICIELS DE COMPÉTENCE

Pour un guide d'étude, il est nécessaire de spécifier que tel ou tel autre sujet est plus ou moins maîtrisé. Les techniques que nous avons étudiées dans le chapitre précédent permettent d'évaluer le niveau global et de confirmer s'il est ou n'est pas atteint. Il nous faut maintenant chercher des modèles qui fournissent évaluation précise par compétence et non plus simplement globalement. Nous avons, ainsi porté nos efforts sur les modèles linéaires qui permettent de décomposer la performance en un ensemble de compétences distinctes.

Dans ce chapitre nous expliquerons le principe d'alignement de ces modèles linéaires. Dans une deuxième partie, nous présentons les modèles que nous avons étudié. Nous finissons avec les algorithmes que nous avons proposés afin de mieux valider la prédiction de nos modèles linéaires choisis.

3.1 Alignement Item/Compétence, Étudiant/Compétence et Étudiant/item

Afin de réaliser un diagnostic précis et rapide, il est nécessaire d'établir le lien entre Compétence/Étudiant, Étudiant/item et Item/Compétence. Cet alignement permettra respectivement de connaître les compétences que l'étudiant maîtrise, si l'étudiant a répondu correctement à la question ou non et les compétences impliquées pour répondre à chaque question. Ces trois types d'alignements permettent de cerner les difficultés de l'étudiant et de l'aider à les surmonter.

3.1.1 Alignement Étudiant/Compétence

L'alignement entre les étudiants et les compétences, peut se présenter sous la forme d'une matrice de n lignes et de k colonnes, dont les lignes présentent les compétences et les colonnes présentent les étudiants, où une valeur supérieure à zéro indique que l'étudiant maîtrise la compétence. La matrice peut contenir des valeurs positives ou négatives. Ainsi, toute valeur supérieure à zéro indique que la compétence est maîtrisée. Sinon, toute valeur égale ou inférieure à zéro indique que la compétence n'est pas maîtrisée. La matrice \mathbf{S} de la figure 3.1 est un exemple d'alignement entre les étudiants et les compétences. Nous remarquons que l'étudiant e_2 maîtrise les compétences s_1 et s_3 .

3.1.2 Alignement Étudiant/item

L'alignement entre les étudiants et les items, peut se présenter sous la forme d'une matrice de m lignes et de k colonnes, dont les lignes présentent les items et les colonnes présentent les étudiants, où une valeur positive indique que l'étudiant a réussi l'item. La matrice peut contenir des valeurs positives ou négatives. Ainsi, toute valeur supérieure à zéro indique que l'étudiant a répondu correctement à l'item. Sinon, toute valeur égale ou inférieure à zéro indique que l'étudiant a échoué l'item. La matrice \mathbf{R} de la figure 3.1 est un exemple de résultats de tests d'étudiants. Nous remarquons que l'étudiant e_2 a réussi les items i_2 , i_3 et i_4 , mais il a échoué l'item i_1 . En effet, la matrice \mathbf{R} est le produit des deux matrices \mathbf{S} et \mathbf{Q} .

Il est clair, d'après l'exemple des matrices d'alignements (figure 3.1), qu'il est un peu difficile de comprendre ces liens surtout avec des valeurs négatives. Ces valeurs peuvent amener à une mauvaise interprétation, surtout si on les présente à un non-expert du domaine.

3.1.3 Alignement Item/Compétence : matrice \mathbf{Q}

L'alignement entre les items et les compétences, peut se présenter sous la forme d'une matrice de m lignes et de n colonnes, dont les lignes présentent les items et les colonnes présentent les compétences requises, où une valeur non nulle indique que la compétence est impliquée dans la réponse à la question. Cette matrice s'appelle la matrice \mathbf{Q} .

La matrice peut contenir des valeurs positives ou négatives. Ainsi, toute valeur supérieure à zéro indique que la compétence est requise pour répondre à la question. Sinon, toute valeur égale ou inférieure à zéro indique que la compétence n'est pas requise.

La matrice \mathbf{Q} de la figure 3.1, est un exemple d'alignement entre les items et les compétences, nous remarquons que pour l'item i_1 on a les compétences s_1 et s_2 qui sont impliquées, puisque les deux valeurs sont supérieures à zéro, alors que la compétence s_3 elle n'est pas requise vu qu'il y a une valeur de -1.

La matrice \mathbf{Q} peut être obtenue par plusieurs méthodes. En effet, elle peut être faite par un expert, ou être dérivée en utilisant une approche statistique en se basant sur les réponses données par les étudiants à un ensemble des questions. Plusieurs approches sta-

$$\begin{array}{c}
 \begin{array}{c} \text{compétences} \\ s_1 \quad s_2 \quad s_3 \end{array} \\
 \mathbf{R} = \begin{array}{c} \text{items} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \begin{array}{|c|c|c|} \hline 1 & \frac{1}{3} & -1 \\ \hline 2 & -2 & 1 \\ \hline 1 & 0 & \frac{1}{2} \\ \hline -1 & 0 & 3 \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{c} \text{étudiants} \\ e_1 \quad e_2 \quad e_3 \end{array} \\
 \mathbf{S} = \begin{array}{c} \text{compétences} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{|c|c|c|} \hline -1 & 2 & 1 \\ \hline \frac{1}{2} & 0 & -1 \\ \hline 0 & 3 & -1 \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{c} \text{étudiants} \\ e_1 \quad e_2 \quad e_3 \end{array} \\
 \mathbf{Q} = \begin{array}{c} \text{items} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \begin{array}{|c|c|c|} \hline \frac{-5}{6} & -1 & \frac{5}{6} \\ \hline -3 & 7 & 3 \\ \hline -1 & \frac{7}{2} & \frac{1}{2} \\ \hline 1 & 7 & -4 \\ \hline \end{array}
 \end{array}$$

Figure 3.1 Exemple de matrices d'alignements.

tistiques existent afin de dériver la matrice \mathbf{Q} à partir de données, dont nous trouvons la méthode de factorisation des matrices en valeurs non négatives.

3.2 Factorisation des matrices en valeurs non négatives

La décomposition des matrices peut conduire à une mauvaise interprétation des données. Cette fausse interprétation nous amène à utiliser une méthode de factorisation de matrice avec des valeurs non négatives. La méthode s'appelle NMF (Non-negative Matrix Factorization) (voir figure 3.2). Elle consiste à décomposer une matrice $\mathbf{R}_{(n \times m)}$ en deux sous matrices plus petites \mathbf{Q} et \mathbf{S} , basé sur les facteurs latents. Dans le cadre de la prédiction des performances des étudiants, ces facteurs latents sont les compétences.

$$\mathbf{R} \approx \mathbf{Q} \mathbf{S} \quad (3.1)$$

La taille des matrices \mathbf{Q} et \mathbf{S} sont respectivement : $n \times r$ et $r \times m$, r s'appelle le rang de factorisation. La matrice \mathbf{R} représente les résultats des n étudiants et leurs succès/échec aux m questions. La matrice \mathbf{Q} correspond à la Q-matrice. Elle représente l'alignement entre les questions et les compétences requises. Enfin, la matrice \mathbf{S} représente l'alignement entre les étudiants et leurs compétences. Ainsi, le produit des deux matrices \mathbf{Q} et \mathbf{S} nous permet d'avoir la matrice contenant le taux de succès/échec des réponses de n étudiants aux m questions.

Les trois matrices ne contiennent que des valeurs positives ou nulles (Figure 3.2). Cette technique donne plus de sens en terme de comprendre l'alignement entre les étudiants/Questions, étudiant/Compétences ou Questions/Compétences.

Prenons l'exemple de la matrice \mathbf{Q} de la figure 3.2, l'interprétation de cette matrice est simple à comprendre, en effet, pour que l'étudiant réussisse la première question i_1 , il a besoin de maîtriser les compétences s_2 et s_3 , et pour i_2 il a besoin seulement de s_3 . Pour la matrice \mathbf{S} , on remarque que l'étudiant e_1 il maîtrise s_1 seulement et l'étudiant e_3 , il maîtrise s_2 et s_3 .

$$\mathbf{Q} = \begin{array}{c} \text{items} \\ \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \end{array} \begin{array}{c} \text{compétences} \\ \begin{array}{c} s_1 \quad s_2 \quad s_3 \end{array} \end{array} \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \quad \mathbf{S} = \begin{array}{c} \text{compétences} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \end{array} \begin{array}{c} \text{étudiants} \\ \begin{array}{c} e_1 \quad e_2 \quad e_3 \end{array} \end{array} \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline \end{array} \quad \mathbf{R} = \begin{array}{c} \text{items} \\ \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \end{array} \begin{array}{c} \text{étudiants} \\ \begin{array}{c} e_1 \quad e_2 \quad e_3 \end{array} \end{array} \begin{array}{|c|c|c|} \hline 0 & 0 & 2 \\ \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & 2 \\ \hline \end{array}$$

Figure 3.2 Exemple des matrices V, W et H.

En ayant ces données, on peut prédire que l'étudiant e_1 va échouer la première question i_1 et va répondre correctement à i_3 . Les résultats sont contenus dans la matrice **R**.

3.3 Modèles d'alignement des questions et des compétences

La dérivation de la matrice **Q** n'est qu'une étape, parmi les autres étapes, afin de pouvoir établir un diagnostic des compétences efficace. En effet, il est nécessaire d'adopter un modèle d'alignement afin d'interpréter cette matrice **Q**. Plusieurs modèles existent, dont nous citons :

- Le modèle conjonctif : chaque compétence est nécessaire pour réussir l'item et l'absence d'une seule compétence résulte l'échec ;
- Le modèle disjonctif : n'importe quelle compétence est suffisante pour réussir l'item ;
- Le modèle compensatoire : chaque compétence augmente la probabilité de succès de répondre à l'item.

Dans notre projet nous avons choisi de travailler avec le modèle compensatoire et le modèle conjonctif.

3.3.1 Le modèle compensatoire

Comme nous avons déjà expliqué ci-dessus, le modèle compensatoire réside sur le principe que chaque compétence augmente la probabilité de succès de répondre correctement à la question. En effet, l'interprétation compensatoire de la modélisation des compétences est représentée par l'équation 3.2

$$\mathbf{R} \approx \mathbf{Q} \mathbf{S} \quad (3.2)$$

Sachant que :

- La matrice **R** représente les résultats des étudiants aux questions (voir section 3.1.2).
- La matrice **S** représente l'alignement entre les étudiants et les compétences (voir section 3.1.1).
- La matrice **Q** représente l'alignement entre les questions et les compétences (voir section 3.1.3).

La matrice **Q** de la figure 3.3 représente un exemple d'alignement entre les questions et les compétences. En effet, avec le modèle compensatoire, chaque compétence s_1 ou s_2 augmente la chance pour que l'étudiant puisse répondre correctement à la question i_1 .

$$\mathbf{Q} = \begin{array}{c} \text{items} \\ \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \end{array} \begin{array}{c} \text{compétences} \\ \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \end{array} \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$$
Figure 3.3 Exemple d'une matrice \mathbf{Q}

3.3.2 Modèle conjonctif

Le modèle conjonctif exige que toutes les compétences requises soient maîtrisées pour répondre à la question. L'équation 3.2 correspond ainsi dans le modèle conjonctif à l'équation 3.3

$$\neg \mathbf{R} = \mathbf{Q}(\neg S) \quad (3.3)$$

Sachant que l'opérateur \neg est l'opérateur booléen de négation, qui est défini comme la fonction qui transforme toute valeur de 0 à 1 et toutes les autres valeurs à 0.

Dans la perspective où l'on tente de prédire les résultats au test, l'équation 3.3 permettra d'attribuer la valeur 0 dans la matrice \mathbf{R} si jamais une ou plusieurs compétences sont absentes et 1 si toutes les compétences requises existent.

Prenons le même exemple de matrice de la figure 3.3. Avec le modèle conjonctif, pour que l'étudiant puisse répondre correctement à la question i_1 , il doit maîtriser les compétences s_2 et s_3 .

3.4 Factorisation et régression linéaire

Dans le but de valider dans quelle mesure les modèles conjonctif et compensatoire sont des représentations valides des compétences, nous avons défini différents algorithmes permettant d'effectuer des prédictions à partir de ces modèles. Nous étudions notamment la capacité de prédire les résultats d'un étudiant donné à partir d'une partie de ses réponses observées. Le principe est que les résultats observés permettent d'estimer les compétences et l'alignement des questions aux compétences. Ces mêmes informations sont en retour utilisées pour prédire les résultats non observés.

Nous avons eu recours à la régression linéaire en utilisant la méthode des moindres carrés afin d'estimer les valeurs des matrices \mathbf{S} et \mathbf{Q} à condition d'avoir la matrice \mathbf{R} avec \mathbf{Q} ou \mathbf{S} (équation 3.4 et équation 3.5) :

$$\hat{\mathbf{S}} = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T \mathbf{R} \quad (3.4)$$

et

$$\hat{\mathbf{Q}} = \mathbf{R} \mathbf{S}^T (\mathbf{S} \mathbf{S}^T)^{-1} \quad (3.5)$$

L'équation 3.4 représente l'estimation de la matrice \mathbf{S} en appliquant la régression linéaire sur les matrices \mathbf{R} et \mathbf{Q} . L'équation 3.5 suit le même principe afin d'estimer \mathbf{Q} . Les matrices estimées avec la méthode des moindres carrés sont représentées par $\hat{\mathbf{S}}$ et $\hat{\mathbf{Q}}$. Les deux équations 3.4 et 3.5 sont la base de la régression dans les algorithmes définis dans les modèles conjonctif et compensatoire que nous allons détailler dans les sections suivantes.

Tous les algorithmes que nous avons utilisés suivent le même principe de prédiction, qui consiste à estimer la matrice $\hat{\mathbf{Q}}$ à partir d'une partie des données. Puis, on utilise le reste des données pour effectuer une validation croisée qui consiste, elle, à estimer, pour chaque étudiant, les compétences avec un sous-ensemble des questions et à prédire le reste des questions sur la base de l'équation (3.4). Finalement, on mesure la proportion de bonnes prédictions par l'équation (3.2) ou l'équation (3.3) en utilisant les matrices $\hat{\mathbf{Q}}$ et $\hat{\mathbf{S}}$. Les matrices $\hat{\mathbf{S}}$ et $\hat{\mathbf{Q}}$ doivent être dérivées d'un sous-ensemble de données indépendant afin de maintenir la validité. Nous avons donc divisé nos données en 4 sous-ensembles (voir figure 3.4).

Tout d'abord, définissons :

- m : nombre de questions
- n : nombre d'étudiants
- k : nombre de compétences

Ainsi, les quatre sous-ensembles des données sont définis comme suit :

- $\mathbf{R}_{(n_1, m_1)}$: les données d'apprentissage
- $\mathbf{R}_{(n_1, m_2)}$: les données de test

		Entrainement	Test
		n_1	n_2
Observation	m_1	$\mathbf{R}_{(m_1, n_1)}$	$\mathbf{R}_{(m_1, n_2)}$
Estimation	m_2	$\mathbf{R}_{(m_2, n_1)}$	$\mathbf{R}_{(m_2, n_2)}$

Figure 3.4 Décomposition des données

- $\mathbf{R}_{(n_2, m_1)}$: les données observées
- $\mathbf{R}_{(n_2, m_2)}$: les données déduites

avec :

- $m = m_1 + m_2$
- $n = n_1 + n_2$

Nous utilisons la notation $\mathbf{A}_{(m, n)}$ pour référer à la dimension (m,n) de la matrice \mathbf{A} . Toutes les données d'entraînement peuvent être représentées par la matrice $\mathbf{R}_{(m, n_1)}$.

Nous expliquerons dans les sections suivantes les algorithmes que nous avons utilisés afin de valider la prédiction de nos deux modèles conjonctif et compensatoire.

3.5 Algorithmes proposés

3.5.1 Algorithme 1

Le premier algorithme est basé sur l'équation suivante :

$$\hat{\mathbf{R}}_{(m_2, n_2)} = \hat{\mathbf{Q}}_{(m_2, k)} \hat{\mathbf{S}}_{(k, n_2)} \quad (3.6)$$

La matrice $\hat{\mathbf{R}}_{(m_2, n_2)}$ est calculée à partir de l'estimation des deux matrices $\hat{\mathbf{Q}}_{(m_2, k)}$ et $\hat{\mathbf{S}}_{(k, n_2)}$. L'estimation de la matrice $\hat{\mathbf{Q}}_{(m_2, k)}$ est obtenue à partir des données test $\mathbf{R}_{(m_2, n_1)}$ de l'ensemble des données d'entraînement. La matrice $\hat{\mathbf{S}}_{(k, n_2)}$ est obtenue de manière similaire, à partir des données de test $\mathbf{R}_{(m_1, n_2)}$ de l'ensemble des données d'entraînement. Les deux estimations suivent le principe de la régression linéaire expliqué par les deux équations (3.4, 3.5). D'un autre côté, dans le but que l'alignement des deux matrices $\hat{\mathbf{Q}}_{(m_2, k)}$ et $\hat{\mathbf{S}}_{(k, n_2)}$ soit fait pour les mêmes facteurs latents, nous utilisons les matrices $\mathbf{Q}_{(m_1, k)}$ et $\mathbf{S}_{(k, n_1)}$ que nous dérivons en utilisant la factorisation NMF sur les données de test :

$$\mathbf{R}_{(m_1, n_1)} \approx \mathbf{Q}_{(m_1, k)} \mathbf{S}_{(k, n_1)} \quad (3.7)$$

Nous détaillons dans ce qui suit le processus d'estimation déjà décrit ci-dessus. Ainsi, définissons la factorisation de $\mathbf{R}_{(m_2, n_1)}$ comme étant :

$$\mathbf{R}_{(m_2, n_1)} \approx \hat{\mathbf{Q}}_{(m_2, k)} \mathbf{S}_{(k, n_1)} \quad (3.8)$$

Le but est d'isoler $\hat{\mathbf{Q}}_{(m_2, k)}$ dans l'équation 3.8 en suivant le principe d'estimation par les moindres carrés de l'équation 3.5, ce qui nous donne l'équation 3.9

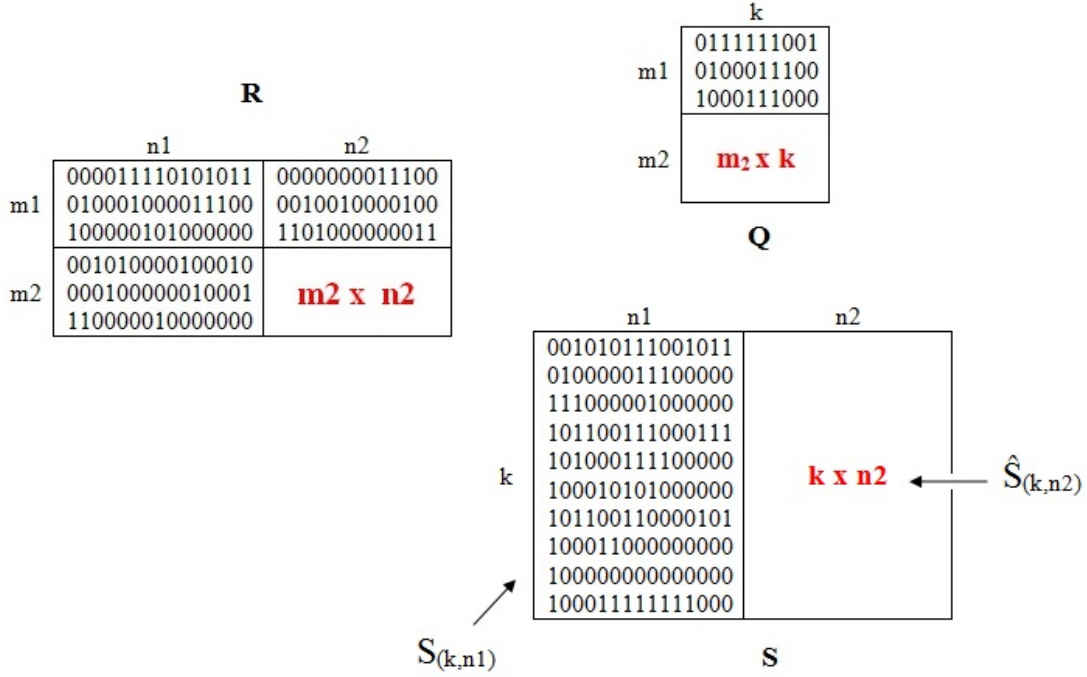


Figure 3.5 Explication du principe de séparation des données

$$\hat{Q}_{(m_2,k)} = R_{(m_2,n_1)} S_{(k,n_1)}^T (S_{(k,n_1)} S_{(k,n_1)}^T)^{-1} \quad (3.9)$$

Le même principe sera appliqué afin de dériver la matrice $\hat{S}_{(k,n_2)}$ à partir des données d'entraînement $R_{(m_1,n_2)}$ qui sont définies comme suit :

$$R_{(m_1,n_2)} \approx Q_{(m_1,k)} \hat{S}_{(k,n_2)} \quad (3.10)$$

Donc, pour isoler la matrice

$$\hat{S}_{(k,n_2)} = (Q_{(m_1,k)}^T Q_{(m_1,k)})^{-1} Q_{(m_1,k)}^T R_{(m_1,n_2)} \quad (3.11)$$

Après l'estimation des deux matrices $\hat{Q}_{(m_2,k)}$ et $\hat{S}_{(k,n_2)}$, il est maintenant possible d'appliquer l'équation (3.6) afin d'estimer les résultats $\hat{R}_{(m_2,n_2)}$. Le calcul de la précision des estimations peut être fait en les comparant aux valeurs observées :

$$\text{tauxd'erreur} = \frac{\|\mathbf{R}_{(m_2, n_2)} - \hat{\mathbf{Q}}_{(m_2, k)} \hat{\mathbf{S}}_{(k, n_2)}\|^2}{(m_2 \times n_2)} \quad (3.12)$$

$$= \frac{\|\mathbf{R}_{(m_2, n_2)} - \hat{\mathbf{R}}_{(m_2, n_2)}\|^2}{(m_2 \times n_2)} \quad (3.13)$$

3.5.2 Algorithme 2

Le deuxième algorithme repose sur le principe d'estimer, à partir des données d'entraînement, comment passer des items observés à une estimation de compétences, puis de passer de ces compétences vers la prédiction des items à prédire. Ceci se traduit parce que l'on nommera une matrice de projection des résultats aux items observés vers des compétences, puis des compétences vers les prédictions par une matrice-Q obtenue par factorisation des données d'entraînement. Ce processus est formalisé dans ce qui suit.

Le deuxième algorithme repose donc premièrement sur une factorisation des items à prédire (m_2) avec les données des étudiants d'entraînement (n_1) :

$$\mathbf{R}_{(m_2, n_1)} \approx \mathbf{Q}_{(m_2, k)} \mathbf{S}_{(k, n_1)} \quad (3.14)$$

Cette factorisation définit, d'une part, une matrice d'alignement de k compétences aux m_2 items à prédire et, d'autre part, assigne ces compétences aux n_1 étudiants d'entraînement. Ensuite, le principe consiste à déterminer une projection des items observés, m_1 , vers ces mêmes compétences :

$$\mathbf{S}_{(k, n_1)} = \mathbf{P}_{(k, m_1)} \mathbf{R}_{(m_1, n_1)} \quad (3.15)$$

En d'autres termes, la matrice $\mathbf{P}_{(k, m_1)}$ doit définir comment les résultats aux items m_1 se traduisent en termes des k compétences obtenues par la factorisation en (3.14). Il s'agit en quelque sorte d'une projection des résultats aux items dans l'espace des compétences de la factorisation. Une approximation de cette matrice de projection est obtenue par la méthode des moindres carrés :

$$\hat{\mathbf{P}}_{(k, m_1)} = \mathbf{S}_{(k, n_1)} \mathbf{R}_{(m_1, n_1)}^T (\mathbf{R}_{(m_1, n_1)} \mathbf{R}_{(m_1, n_1)}^T)^{-1} \quad (3.16)$$

La matrice $\hat{\mathbf{P}}_{(k, m_1)}$ permet par la suite d'estimer les compétences de l'ensemble des étudiants de test, n_2 , à partir de leur réponse aux items observés, m_1 :

$$\hat{\mathbf{S}}_{(k, n_2)} = \hat{\mathbf{P}}_{(k, m_1)} \mathbf{R}_{(m_1, n_2)} \quad (3.17)$$

Et finalement d'estimer le succès aux items non observés, m_2 , à partir de cette estimation des compétences :

$$\hat{\mathbf{R}}_{(m_2, n_2)} = \mathbf{Q}_{(m_2, k)} \hat{\mathbf{S}}_{(k, n_2)} \quad (3.18)$$

3.5.3 Algorithme 3

Le troisième algorithme réutilise la notion de projection, mais cette fois pour passer directement des items observés aux items à prédire :

$$\mathbf{R}_{(m_2, n_1)} \approx \mathbf{P}_{(m_2, m_1)} \mathbf{R}_{(m_1, n_1)} \quad (3.19)$$

Et tout comme pour l'algorithme 3, la matrice de projection, $\mathbf{P}_{(m_2, m_1)}$, est estimée par les moindres carrés et en utilisant les données d'entraînement $\mathbf{R}_{(m, n_1)}$:

$$\hat{\mathbf{P}}_{(m_2, m_1)} \approx \mathbf{R}_{(m_2, n_1)} \mathbf{R}_{(m_1, n_1)}^T (\mathbf{R}_{(m_1, n_1)} \mathbf{R}_{(m_1, n_1)}^T)^{-1} \quad (3.20)$$

La matrice de projection estimée, $\hat{\mathbf{P}}_{(m_2, m_1)}$, est ensuite réutilisée avec les données de test, n_2 :

$$\hat{\mathbf{R}}_{(m_2, n_2)} \approx \hat{\mathbf{P}}_{(m_2, m_1)} \mathbf{R}_{(m_1, n_2)} \quad (3.21)$$

3.5.4 Algorithme 4

Le quatrième algorithme utilise deux ensembles de facteurs latents. Les facteurs latents des données d'entraînement et des données de test. Pour ce faire, on dérive les facteurs latents en appliquant deux factorisations NMF et on définit un modèle de régression, afin de prédire les facteurs latents des données prédites à partir des facteurs latents des données d'observation. Le processus de prédiction est le suivant :

Les deux factorisations sont appliquées respectivement sur les données observées et les données prédites de l'ensemble de données de test :

$$\mathbf{R}_{(m_1, n_1)} \approx \mathbf{Q}_{(m_1, k)} \mathbf{S}_1_{(k, n_1)} \quad (3.22)$$

$$\mathbf{R}_{(m_2, n_1)} \approx \mathbf{Q}_{(m_2, k)} \mathbf{S}_2_{(k, n_1)} \quad (3.23)$$

Ensuite, une matrice de projection est obtenue afin de dériver la matrice $\mathbf{S}_{2(k,n_1)}$ de la matrice $\mathbf{S}_{1(k,n_1)}$

$$\mathbf{S}_{2(k,n_1)} = \hat{\mathbf{P}}_{(k,k)} \mathbf{S}_{1(k,n_1)} \quad (3.24)$$

$$\mathbf{P}_{(k,k)} = \mathbf{S}_{2(k,n_1)} \mathbf{S}_{1(k,n_1)}^T (\mathbf{S}_{1(k,n_1)} \mathbf{S}_{1(k,n_1)}^T)^{-1} \quad (3.25)$$

Une estimation des compétences des étudiants de l'ensemble de test, est obtenue en se basant sur l'équation (3.22) :

$$\hat{\mathbf{S}}_{1(k,n_2)} = (\mathbf{Q}_{(m_1,k)}^T \mathbf{Q}_{(m_1,k)})^{-1} \mathbf{Q}_{(m_1,k)}^T \mathbf{R}_{(m_1,n_2)} \quad (3.26)$$

$$(3.27)$$

En utilisant la matrice de projection de l'équation (3.25), $\mathbf{P}_{(k,k)}$, on obtient les compétences des étudiants de l'ensemble de test :

$$\hat{\mathbf{S}}_{2(k,n_2)} = \mathbf{P}_{(k,k)} \hat{\mathbf{S}}_{1(k,n_2)} \quad (3.28)$$

Finalement les résultats de prédiction sont :

$$\hat{\mathbf{R}}_{(m_2,n_2)} = \mathbf{Q}_{(m_2,k)} \hat{\mathbf{S}}_{2(k,n_2)} \quad (3.29)$$

Il est à noter que ce modèle nécessite deux étapes de factorisation (équations 3.22 et 3.23).

3.5.5 IRT et modèle des valeurs attendues

Pour des fins de validation et de comparaison, nous avons inclus les deux modèles de IRT et celui des valeurs attendues.

Comme nous avons déjà expliqué dans la section 2.4.2, le modèle IRT est très utilisé pour l'évaluation des compétences. Le modèle IRT est un modèle de régression logistique à deux paramètres représentant respectivement la difficulté d'un item et sa discrimination. C'est

un modèle à compétence unique. Les compétences des étudiants sont dérivées à partir des données d'observation qui sont utilisées afin de prédire les données non-observées.

L'autre modèle utilisé est celui des valeurs attendues, qui correspond aux résultats attendus en se basant sur la moyenne géométrique du taux de succès moyen de réponse des étudiants et du taux de succès moyen des questions. Le taux de succès des items non observés, est basé sur $\mathbf{R}_{(m_2, n_1)}$ et le taux du succès des étudiants est estimé à partir de $\mathbf{R}_{(m_1, n_2)}$.

En définissant le vecteur colonne \mathbf{u} comme le taux de succès de chaque item et le vecteur ligne \mathbf{v} comme le taux de succès de chaque étudiant, on obtient une matrice par le produit des deux. La racine carrée de cette matrice représente la moyenne géométrique, qui est la valeur attendue à chaque item et pour chaque sujet par : $\sqrt{\mathbf{u}\mathbf{v}}$

À partir des deux modèles IRT et valeurs attendues, nous avons conçu deux autres algorithmes qui sont l'algorithme 5 et l'algorithme 6

3.5.6 Algorithme 5

L'algorithme 5 permet de générer une matrice \mathbf{R}_2 qui est la combinaison entre les données générées avec l'IRT et celui des données d'entraînement (voir figure 3.6). Ainsi à partir des matrices \mathbf{R} (matrice des données originale) et

$$\mathbf{R}_{2(m_1, n_1)} = \mathbf{R}_{(m_1, n_1)} \quad (3.30)$$

$$\mathbf{R}_{2(m_2, n_1)} = \mathbf{R}_{(m_2, n_1)} \quad (3.31)$$

$$\mathbf{R}_{2(m_1, n_2)} = \mathbf{R}_{(m_1, n_2)} \quad (3.32)$$

$$\mathbf{R}_{2(m_2, n_2)} = \mathbf{R}_{\text{IRT}(m_2, n_2)} \quad (3.33)$$

Ainsi, la matrice \mathbf{R}_2 s'écrit :

$$\mathbf{R}_2 \approx \mathbf{Q}_2 \mathbf{S}_2 \quad (3.34)$$

À partir de \mathbf{R}_2 nous allons dériver les matrices $\hat{\mathbf{Q}}_2$ et $\hat{\mathbf{S}}_2$ en utilisant la méthode des moindres carrés :

$$\hat{\mathbf{Q}}_2 = \mathbf{R}_2 \mathbf{S}_2^T (\mathbf{S}_2 \mathbf{S}_2^T)^{-1} \quad (3.35)$$

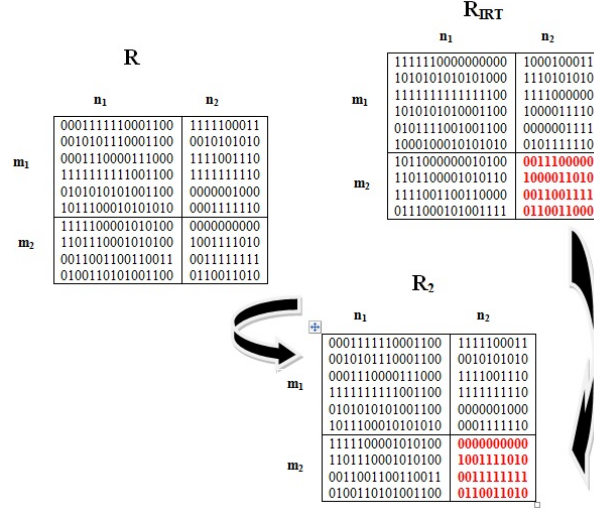


Figure 3.6 Principe de génération de la matrice des données de l'algorithme 5

$$\hat{\mathbf{S}}_2 = (\mathbf{Q}_2^T \mathbf{Q}_2)^{-1} \mathbf{Q}_2^T \mathbf{R}_2 \quad (3.36)$$

L'estimation est faite en faisant le produit des matrices factorisées $\hat{\mathbf{Q}}_2$ et $\hat{\mathbf{S}}_2$, nous obtenons ainsi la matrice $\hat{\mathbf{R}}_2$:

$$\hat{\mathbf{R}}_2 = \hat{\mathbf{Q}}_2 \hat{\mathbf{S}}_2 \quad (3.37)$$

3.5.7 Algorithme 6

L'algorithme 6 suit le même principe que celui de l'algorithme 5, en prenant cette fois-ci la matrice générée avec le modèle des valeurs attendues $\mathbf{R}_{\text{val-att}}$ et la matrice des données d'entraînement \mathbf{R} . Cela nous a permis de générer une matrice \mathbf{R}_3 (voir figure 3.7) :

$$\mathbf{R}_{\mathbf{3}(m_1, n_1)} = \mathbf{R}_{(m_1, n_1)} \quad (3.38)$$

$$\mathbf{R}_{\mathbf{3}(m_2, n_1)} = \mathbf{R}_{(m_2, n_1)} \quad (3.39)$$

$$\mathbf{R}_{\mathbf{3}(m_1, n_2)} = \mathbf{R}_{(m_1, n_2)} \quad (3.40)$$

$$\mathbf{R}_{\mathbf{3}(m_2, n_2)} = \mathbf{R}_{\text{val-att}(m_2, n_2)} \quad (3.41)$$

Ainsi, la matrice \mathbf{R}_3 s'écrit :

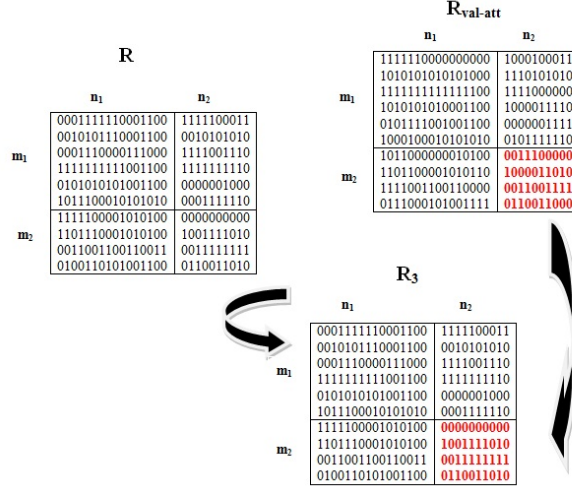


Figure 3.7 Principe de génération de la matrice des données de l'algorithme 6

$$\mathbf{R}_3 \approx \mathbf{Q}_3 \mathbf{S}_3 \quad (3.42)$$

À partir de \mathbf{R}_3 nous allons dériver les matrices $\hat{\mathbf{Q}}_3$ et $\hat{\mathbf{S}}_3$ en utilisant la méthode des moindres carrés :

$$\hat{\mathbf{Q}}_3 = \mathbf{R}_3 \mathbf{S}_3^T (\mathbf{S}_3 \mathbf{S}_3^T)^{-1} \quad (3.43)$$

$$\hat{\mathbf{S}}_3 = (\mathbf{Q}_3^T \mathbf{Q}_3)^{-1} \mathbf{Q}_3^T \mathbf{R}_3 \quad (3.44)$$

L'estimation est faite en faisant le produit des matrices factorisées $\hat{\mathbf{Q}}_3$ et $\hat{\mathbf{S}}_3$, nous obtenons ainsi la matrice $\hat{\mathbf{R}}_3$:

$$\hat{\mathbf{R}}_3 = \hat{\mathbf{Q}}_3 \hat{\mathbf{S}}_3 \quad (3.45)$$

CHAPITRE 4

RÉSULTATS ET ANALYSES

Les modèles définis dans le chapitre précédent constituent une alternative aux modèles étudiés dans la revue de littérature. Il reste maintenant à les évaluer sur la base de la précision des prédictions qu'on peut effectuer avec des données réelles.

4.1 Présentation des données

Afin de valider nos algorithmes, nous avons utilisé trois ensembles de données différents, issus de questionnaires réellement soumis à des étudiants :

1. Test d'arithmétique (Vomlel) : ces données proviennent d'un test d'arithmétique des fractions et comporte 20 questions qui ont été posées à 149 élèves de 10 à 12 ans. La source de ces données est [Vomlel (2004)]. Le score moyen est de 61 %.
2. Test de commandes du shell Unix : ce jeu de données provient d'un test de 34 questions, donné à 48 étudiants. Les questions portent sur différentes commandes de terminal Unix, allant de commandes simples (exemple : ls, cd), à des commandes plus complexes et des commandes d'administration système. Cet ensemble de données est caractérisé par le fait que le nombre d'étudiants est petit et la performance moyenne est de 54%, mais aussi par une variance élevée avec une distribution rectangulaire des scores des étudiants.
3. Test de mathématiques de Polytechnique (Polytechnique) : ce jeu de test est composé de 60 questions et soumis à 250 étudiants en première année de baccalauréat de l'école Polytechnique de Montréal. Le score moyen est de 58%.

Le tableau 4.1 résume le taux de réussite moyen de l'ensemble des élèves pour chaque questionnaire.

Tableau 4.1 Résumé des jeux de données

Jeu de données	Nb. de question	Nb. de répondants	Taux de réussite moyen
Vomlel	160	41	61%
Unix	34	48	54%
Polytechnique	60	250	58%

4.2 D roulement des simulations

Afin de valider la performance des deux mod les conjonctif et compensatoire, nous avons effectu  un ensemble de tests. Les conditions et les param tres sont les suivants :

- Nous avons pris diff rentes valeurs de facteurs latents (comp tences). Dans cette section nous pr sentons les r sultats obtenus avec un nombre de facteurs latents ($n.skills$) = 5. Vous pouvez vous r f rer   l’annexe B afin de voir d’autres r sultats avec d’autres choix du nombre de comp tences.
- Pour tous les algorithmes o  on utilise le NMF pour la factorisation, on r p te la factorisation 6 fois afin d’avoir une estimation moyenne des matrices \hat{Q} et \hat{S}
- Chaque exp rience est r p t e 24 fois afin d’avoir une pr diction moyenne.
- Tous les mod les donnent les r sultats de pr diction pour les n_2 questions non observ es (donn es de test), en se basant sur les n_1 questions observ es.
- Le calcul de la pr cision est fait en consid rant le taux du succ s de la pr diction des items. Toutes les donn es r sultantes dans toutes les matrices estim es \hat{R} sont arrondies   $\{0,1\}$.
- pour le mod le IRT, nous avons utilis  la fonction `ltm` dans le cadre de R [Rizopoulos (2006)].

4.3 R sultats du mod le compensatoire

Dans cette section nous vous pr sentons les r sultats obtenus pour le mod le compensatoire.

4.3.1 R sultats obtenus avec les donn es de Vomlel

Les r sultats obtenus avec les donn es de Vomlel sont pr sent s par l’ensemble des tableaux 4.2, 4.3 et 4.4.

Le tableau 4.2 illustre les r sultats obtenus pour chaque algorithme avec le mod le compensatoire, en utilisant les donn es de Vomlel avec les param tres suivants :

- nombre de comp tences ($n.skills$)= 5
- nombre de donn es observ es (nombre de questions observ es) ($obs.size$) = 5
- donn es d’entrainement ($train.size$) = 128
- nombre de r p titions ($n.run$) = 24

Le tableau 4.2 pr sente les taux d’erreur obtenus pour chaque algorithme. Chaque valeur repr sente la moyenne des 24 r p titions et l’ cart-type est calcul  pour ces r p titions.

Le tableau 4.3 illustre les r sultats obtenus pour chaque algorithme avec le mod le compensatoire, en utilisant les donn es de Vomlel et avec les m mes param tres que la section

Tableau 4.2 Tableau des résultats de performance avec les données de Vomlel du modèle compensatoire, n.skills = 5 et obs.size = 5.

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	128	5	0.228	0.037
Algorithme 2	5	24	128	5	0.227	0.038
Algorithme 3	5	24	128	5	0.227	0.036
Algorithme 4	5	24	128	5	0.228	0.037
Algorithme 5	5	24	128	5	0.158	0.026
Algorithme 6	5	24	128	5	0.235	0.039
Modèle IRT	5	24	128	5	0.155	0.023
Modèle valeurs attendues	5	24	128	5	0.248	0.042

précédente (4.3.1), mais cette fois en fixant le nombre de données (questions) observées à 10.

Le tableau 4.3 présente les taux d'erreur obtenus pour chaque algorithme.

Le tableau 4.4 rapporte les résultats obtenus pour chaque algorithme avec le modèle compensatoire, en utilisant les données de Vomlel et avec les mêmes paramètres que la section précédente (4.3.1), mais cette fois en fixant le nombre de données (questions) observées à 15.

Le tableau 4.4 présente les taux d'erreur obtenus pour chaque algorithme.

4.3.2 Résultats obtenus avec les données d'Unix

Les résultats obtenus avec les données d'Unix sont présentés par l'ensemble des tableaux 4.5, 4.6 et 4.7.

Le tableau 4.5 illustre les résultats obtenus pour chaque algorithme avec le modèle compensatoire, en utilisant les données d'Unix avec les paramètres suivants :

- nombre de compétences (n.skills)= 5
- nombre de données observées (nombre de questions observées) (obs.size) = 10
- données d'entraînement (train.size) = 40
- nombre de répétitions (n.run) = 24

Le tableau 4.6 illustre les résultats obtenus pour chaque algorithme avec le modèle compensatoire, en utilisant les données d'Unix et avec les mêmes paramètres que la section précédente (4.3.2), mais cette fois en fixant le nombre de données (questions) observées à 20.

Le tableau 4.7 illustre les résultats obtenus pour chaque algorithme avec le modèle com-

Tableau 4.3 Tableau des résultats de performance avec les données de Vomlel du modèle compensatoire, n.skills = 5 et obs.size = 10

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	128	10	0.186	0.079
Algorithme 2	5	24	128	10	0.189	0.082
Algorithme 3	5	24	128	10	0.186	0.084
Algorithme 4	5	24	128	10	0.185	0.085
Algorithme 5	5	24	128	10	0.152	0.079
Algorithme 6	5	24	128	10	0.192	0.082
Modèle IRT	5	24	128	10	0.151	0.079
Modèle valeurs attendues	5	24	128	10	0.221	0.080

Tableau 4.4 Tableau des résultats de performance avec les données de Vomlel du modèle compensatoire, n.skills = 5 et obs.size = 15

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	128	15	0.190	0.046
Algorithme 2	5	24	128	15	0.189	0.047
Algorithme 3	5	24	128	15	0.183	0.054
Algorithme 4	5	24	128	15	0.182	0.055
Algorithme 5	5	24	128	15	0.148	0.038
Algorithme 6	5	24	128	15	0.190	0.044
Modèle IRT	5	24	128	15	0.133	0.054
Modèle valeurs attendues	5	24	128	15	0.228	0.047

Tableau 4.5 Tableau des résultats de performance avec les données de Unix du modèle compensatoire, n.skills = 5 et obs.size = 10

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	40	10	0.155	0.031
Algorithme 2	5	24	40	10	0.155	0.034
Algorithme 3	5	24	40	10	0.155	0.036
Algorithme 4	5	24	40	10	0.164	0.038
Algorithme 5	5	24	40	10	0.116	0.031
Algorithme 6	5	24	40	10	0.148	0.045
Modèle IRT	5	24	40	10	0.108	0.033
Modèle valeurs attendues	5	24	40	10	0.161	0.051

Tableau 4.6 Tableau des résultats de performance avec les données de Unix du modèle compensatoire, n.skills = 5 et obs.size = 20

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	40	20	0.146	0.042
Algorithme 2	5	24	40	20	0.142	0.043
Algorithme 3	5	24	40	20	0.212	0.086
Algorithme 4	5	24	40	20	0.265	0.098
Algorithme 5	5	24	40	20	0.123	0.039
Algorithme 6	5	24	40	20	0.149	0.036
Modèle IRT	5	24	40	20	0.106	0.040
Modèle valeurs attendues	5	24	40	20	0.171	0.038

Tableau 4.7 Tableau des résultats de performance avec les données de Unix du modèle compensatoire, n.skills = 5 et obs.size = 25

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	40	25	0.132	0.043
Algorithme 2	5	24	40	25	0.132	0.048
Algorithme 3	5	24	40	25	0.276	0.089
Algorithme 4	5	24	40	25	0.323	0.094
Algorithme 5	5	24	40	25	0.116	0.045
Algorithme 6	5	24	40	25	0.135	0.048
Modèle IRT	5	24	40	25	0.090	0.049
Modèle valeurs attendues	5	24	40	25	0.130	0.055

pensatoire, en utilisant les données d'Unix et avec les mêmes paramètres que la section précédente (4.3.2), mais cette fois en fixant le nombre de données (questions) observées à 25.

4.3.3 Résultats obtenus avec les données de Polytechnique

Les résultats obtenus avec les données de Polytechnique sont présentés par l'ensemble des tableaux 4.8, 4.9 et 4.10.

Le tableau 4.8 illustre les résultats obtenus pour chaque algorithme avec le modèle compensatoire, en utilisant les données de Polytechnique avec les paramètres suivants :

- nombre de compétences = 5
- nombre de données observées (nombre de questions observées) (obs.size) = 15
- données d'entraînement = 200
- nombre de répétitions = 24

Le tableau 4.9 illustre les résultats obtenus pour chaque algorithme avec le modèle compensatoire, en utilisant les données de Polytechnique et avec les mêmes paramètres que la section précédente (4.3.3), mais cette fois en fixant le nombre de données (questions) observées à 30.

Le tableau 4.10 illustre les résultats obtenus pour chaque algorithme avec le modèle compensatoire, en utilisant les données de Polytechnique et avec les mêmes paramètres que la section précédente (4.3.3), mais cette fois en fixant le nombre de données (questions) observées à 45.

4.4 Résultats avec le modèle conjonctif

Dans cette section nous vous présentons les résultats obtenus pour le modèle conjonctif.

4.4.1 Résultats obtenus avec les données de Vomlel

Les résultats obtenus avec les données de Vomlel sont présentés par l'ensemble des tableaux 4.11, 4.12 et 4.13.

Le tableau 4.11 rapporte les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données de Vomlel avec les paramètres suivants :

- nombre de compétences (n.skills)= 5
- nombre de données observées (nombre de questions observées) (obs.size) = 5
- données d'entraînement (train.size) = 128
- nombre de répétitions (n.run) = 24

Tableau 4.8 Tableau des résultats de performance avec les données de Polytechnique du modèle compensatoire, n.skills = 5 et obs.size = 15

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	200	15	0.320	0.014
Algorithme 2	5	24	200	15	0.314	0.015
Algorithme 3	5	24	200	15	0.318	0.016
Algorithme 4	5	24	200	15	0.330	0.014
Algorithme 5	5	24	200	15	0.296	0.013
Algorithme 6	5	24	200	15	0.312	0.020
Modèle IRT	5	24	200	15	0.298	0.015
Modèle valeurs attendues	5	24	200	15	0.314	0.021

Tableau 4.9 Tableau des résultats de performance avec les données de Polytechnique du modèle compensatoire, n.skills = 5 et obs.size = 30

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	200	30	0.302	0.014
Algorithme 2	5	24	200	30	0.303	0.015
Algorithme 3	5	24	200	30	0.307	0.013
Algorithme 4	5	24	200	30	0.327	0.014
Algorithme 5	5	24	200	30	0.291	0.017
Algorithme 6	5	24	200	30	0.303	0.016
Modèle IRT	5	24	200	30	0.292	0.018
Modèle valeurs attendues	5	24	200	30	0.309	0.020

Tableau 4.10 Tableau des résultats de performance avec les données de Polytechnique du modèle compensatoire, n.skills = 5 et obs.size = 45

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	200	45	0.301	0.029
Algorithme 2	5	24	200	45	0.302	0.028
Algorithme 3	5	24	200	45	0.322	0.027
Algorithme 4	5	24	200	45	0.334	0.027
Algorithme 5	5	24	200	45	0.295	0.028
Algorithme 6	5	24	200	45	0.303	0.029
Modèle IRT	5	24	200	45	0.290	0.027
Modèle valeurs attendues	5	24	200	45	0.304	0.030

Tableau 4.11 Tableau des résultats de performance avec les données de Vomlel du modèle conjonctif, n.skills = 5 et obs.size = 5

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	128	5	0.235	0.036
Algorithme 2	5	24	128	5	0.242	0.033
Algorithme 3	5	24	128	5	0.238	0.032
Algorithme 4	5	24	128	5	0.234	0.037
Algorithme 5	5	24	128	5	0.151	0.022
Algorithme 6	5	24	128	5	0.239	0.037
Modèle IRT	5	24	128	5	0.144	0.023
Modèle valeurs attendues	5	24	128	5	0.247	0.040

Le tableau 4.12 illustre les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données de Vomle et avec les mêmes paramètres que la section précédente (4.4.1), mais cette fois en fixant le nombre de données (questions) observées à 10.

Le tableau 4.13 illustre les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données de Vomle et avec les mêmes paramètres que la section précédente (4.4.1), mais cette fois en fixant le nombre de données (questions) observées à 15.

4.4.2 Résultats obtenus avec les données d'Unix

Les résultats obtenus avec les données d'Unix sont présentés par l'ensemble des tableaux 4.14, 4.15 et 4.16.

Le tableau 4.14 rapporte les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données d'Unix avec les paramètres suivants :

- nombre de compétences (n.skills)= 5
- nombre de données observées (nombre de questions observées)(obs.size) = 10
- données d'entraînement (train.size) = 40
- nombre de répétitions (n.run) = 24

Le tableau 4.15 rapporte les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données d'Unix et avec les mêmes paramètres que la section précédente (4.4.2), mais cette fois en fixant le nombre de données (questions) observées à 20.

Le tableau 4.16 illustre les résultats obtenus pour chaque algorithme avec le modèle

Tableau 4.12 Tableau des résultats de performance avec les données de Vomlel du modèle conjonctif, n.skills = 5 et obs.size = 10

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	128	10	0.207	0.086
Algorithme 2	5	24	128	10	0.206	0.087
Algorithme 3	5	24	128	10	0.208	0.097
Algorithme 4	5	24	128	10	0.207	0.096
Algorithme 5	5	24	128	10	0.164	0.082
Algorithme 6	5	24	128	10	0.212	0.089
Modèle IRT	5	24	128	10	0.154	0.083
Modèle valeurs attendues	5	24	128	10	0.245	0.085

Tableau 4.13 Tableau des résultats de performance avec les données de Vomlel du modèle conjonctif, n.skills = 5 et obs.size = 15

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	128	15	0.196	0.034
Algorithme 2	5	24	128	15	0.195	0.040
Algorithme 3	5	24	128	15	0.198	0.038
Algorithme 4	5	24	128	15	0.198	0.037
Algorithme 5	5	24	128	15	0.161	0.040
Algorithme 6	5	24	128	15	0.193	0.037
Modèle IRT	5	24	128	15	0.130	0.030
Modèle valeurs attendues	5	24	128	15	0.223	0.055

Tableau 4.14 tableau des résultats de performance avec les données de Unix du modèle conjonctif, n.skills = 5 et obs.size = 10

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	40	10	0.149	0.024
Algorithme 2	5	24	40	10	0.150	0.029
Algorithme 3	5	24	40	10	0.156	0.036
Algorithme 4	5	24	40	10	0.166	0.038
Algorithme 5	5	24	40	10	0.125	0.026
Algorithme 6	5	24	40	10	0.151	0.032
Modèle IRT	5	24	40	10	0.118	0.033
Modèle valeurs attendues	5	24	40	10	0.155	0.043

Tableau 4.15 Tableau des résultats de performance avec les données de Unix du modèle conjonctif, n.skills = 5 et obs.size = 20

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	40	20	0.129	0.039
Algorithme 2	5	24	40	20	0.130	0.037
Algorithme 3	5	24	40	20	0.194	0.057
Algorithme 4	5	24	40	20	0.241	0.077
Algorithme 5	5	24	40	20	0.114	0.026
Algorithme 6	5	24	40	20	0.122	0.039
Modèle IRT	5	24	40	20	0.090	0.033
Modèle valeurs attendues	5	24	40	20	0.119	0.046

conjonctif, en utilisant les données d'Unix et avec les mêmes paramètres que la section précédente (4.4.2), mais cette fois en fixant le nombre de données (questions) observées à 25.

4.4.3 Résultats obtenus avec les données de Polytechnique

Les résultats obtenus avec les données de Polytechnique sont présentés par l'ensemble des tableaux 4.17, 4.18 et 4.19.

Le tableau 4.17 illustre les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données d'Unix avec les paramètres suivants :

- nombre de compétences (n.skills)= 5
- nombre de données observées (nombre de questions observées)(obs.size) = 15
- données d'entraînement (train.size) = 200
- nombre de répétitions (n.run) = 24

Le tableau 4.18 illustre les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données de Polytechnique et avec les mêmes paramètres que la section précédente (4.4.3), mais cette fois en fixant le nombre de données (questions) observées à 30.

Le tableau 4.19 illustre les résultats obtenus pour chaque algorithme avec le modèle conjonctif, en utilisant les données de Polytechnique et avec les mêmes paramètres que la section précédente (4.4.3), mais cette fois en fixant le nombre de données (questions) observées à 45.

Tableau 4.16 tableau des résultats de performance avec les données de Unix du modèle conjonctif, n.skills = 5 et obs.size = 25

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	40	25	0.150	0.046
Algorithme 2	5	24	40	25	0.158	0.044
Algorithme 3	5	24	40	25	0.273	0.095
Algorithme 4	5	24	40	25	0.325	0.114
Algorithme 5	5	24	40	25	0.129	0.042
Algorithme 6	5	24	40	25	0.159	0.047
Modèle IRT	5	24	40	25	0.110	0.043
Modèle valeurs attendues	5	24	40	25	0.175	0.046

Tableau 4.17 Tableau des résultats de performance avec les données de Polytechnique du modèle conjonctif, n.skills = 5 et obs.size = 15

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	200	15	0.322	0.017
Algorithme 2	5	24	200	15	0.318	0.017
Algorithme 3	5	24	200	15	0.320	0.017
Algorithme 4	5	24	200	15	0.325	0.014
Algorithme 5	5	24	200	15	0.296	0.015
Algorithme 6	5	24	200	15	0.318	0.016
Modèle IRT	5	24	200	15	0.298	0.014
Modèle valeurs attendues	5	24	200	15	0.314	0.019

Tableau 4.18 tableau des résultats de performance avec les données de Polytechnique du modèle conjonctif, n.skills = 5 et obs.size = 30

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	200	30	0.306	0.023
Algorithme 2	5	24	200	30	0.305	0.021
Algorithme 3	5	24	200	30	0.310	0.021
Algorithme 4	5	24	200	30	0.326	0.021
Algorithme 5	5	24	200	30	0.294	0.019
Algorithme 6	5	24	200	30	0.307	0.022
Modèle IRT	5	24	200	30	0.291	0.018
Modèle valeurs attendues	5	24	200	30	0.307	0.021

Tableau 4.19 Tableau des résultats de performance avec les données de Polytechnique du modèle conjonctif, n.skills = 5 et obs.size = 45

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen	écart-type
Algorithme 1	5	24	200	45	0.307	0.030
Algorithme 2	5	24	200	45	0.307	0.029
Algorithme 3	5	24	200	45	0.323	0.029
Algorithme 4	5	24	200	45	0.343	0.031
Algorithme 5	5	24	200	45	0.300	0.031
Algorithme 6	5	24	200	45	0.313	0.032
Modèle IRT	5	24	200	45	0.289	0.032
Modèle valeurs attendues	5	24	200	45	0.310	0.035

4.5 Analyse des résultats

Après avoir terminé la phase de simulation, nous analysons les résultats des deux modèles, conjonctif et compensatoire. Dans une première partie, nous analysons les résultats obtenus pour chaque modèle des différents tests effectués. Dans une deuxième partie, nous présentons les analyses que nous avons faites à partir de la comparaison des résultats obtenus pour les deux modèles. Nous finissons cette section par une analyse générale de tous les résultats.

4.5.1 Analyses pour les résultats avec le modèle compensatoire

Nous présentons dans cette partie les résultats obtenus pour le modèle compensatoire. Nous comparons les résultats pour chaque jeu de données.

Le tableau 4.20 présente les résultats des taux d'erreur obtenus avec le jeu de données de Vomlel. Nous avons pris différentes tailles de données d'observation (obs.size) soient 5, 10 et 15. Nous remarquons que les algorithmes 1, 2 et le modèle des valeurs attendues, ont donné les meilleurs résultats avec obs.size = 10. Par contre, les autres algorithmes ont donné des meilleurs résultats avec obs.size=15. En ce qui concerne obs.size = 5, nous remarquons qu'aucun a donné des bons résultats, sauf l'algorithme 5 et le IRT. Cela est dû au fait que le nombre de facteurs latents (compétence) est égal au nombre d'observations. L'algorithme 5 est celui qui a donné les meilleurs résultats par rapport à tous les autres algorithmes, à l'exception de IRT qui demeure toujours le plus performant.

Le tableau 4.21 présente les résultats des taux d'erreur obtenus avec le jeu de données d'Unix. Nous avons pris différentes tailles de données d'observation (obs.size) soient 10, 20 et 25. Nous remarquons que les algorithmes 3 et 4, ont donné les meilleurs résultats avec obs.size = 10. Ils ont donné les mauvais résultats avec obs.size = 20 et obs.size=25. Les

Tableau 4.20 Tableau comparatif des résultats de taux d'erreurs pour les données de Vomlel (modèle compensatoire)

Modèles	obs.size=05 (σ)	obs.size=10 (σ)	obs.size=15 (σ)
Algorithme 1	0.228 (0.037)	0.186 (0.079)	0.190 (0.046)
Algorithme 2	0.227 (0.038)	0.189 (0.082)	0.189 (0.047)
Algorithme 3	0.227 (0.036)	0.186 (0.084)	0.183 (0.054)
Algorithme 4	0.228 (0.037)	0.185 (0.085)	0.182 (0.055)
Algorithme 5	0.158 (0.026)	0.152 (0.079)	0.148 (0.038)
Algorithme 6	0.235 (0.039)	0.192 (0.082)	0.190 (0.044)
Modèle IRT	0.155 (0.023)	0.151 (0.079)	0.133 (0.054)
Modèle valeurs attendues	0.248 (0.042)	0.221 (0.080)	0.228 (0.047)

algorithmes 3 et 4 semblent les moins stables avec le jeu de données d'Unix. Le modèle des valeurs attendues a bien performé, aussi bien l'algorithme 6. L'algorithme 5 et encore une fois, est celui qui a le mieux performé par rapport aux algorithmes 1, 2, 3, 4 et 6. Le modèle IRT demeure toujours le meilleur.

Le tableau 4.22 présente les résultats des taux d'erreur obtenus avec le jeu de données de Polytechnique. Nous avons pris différentes valeurs pour la taille de l'ensemble des données d'observation (obs.size) soient 15, 30 et 45. Nous remarquons que les algorithmes 3, 4, 5, 6 donnent les meilleurs résultats avec obs.size = 30. Le reste des algorithmes ont donné des meilleurs résultats avec obs.size=45. Encore une autre fois, l'algorithme 4 est celui le moins stable par rapport aux autres algorithmes. Nous remarquons que l'algorithme 5 et le IRT ont donné les meilleurs résultats et malgré que l'algorithme 5 a donné des résultats nettement mieux que IRT, mais il suffit de vérifier les écarts-type pour comprendre que cette différence est due au bruit.

Nous pouvons confirmer que pour tous les algorithmes, l'IRT est celui qui a donné les

Tableau 4.21 Tableau comparatif des résultats de taux d'erreurs pour les données de Unix (modèle compensatoire)

Modèles	obs.size=10 (σ)	obs.size=20 (σ)	obs.size=25 (σ)
Algorithme 1	0.155 (0.031)	0.146 (0.042)	0.132 (0.043)
Algorithme 2	0.155 (0.034)	0.142 (0.043)	0.132 (0.048)
Algorithme 3	0.155 (0.036)	0.212 (0.086)	0.276 (0.089)
Algorithme 4	0.164 (0.038)	0.265 (0.098)	0.323 (0.094)
Algorithme 5	0.116 (0.031)	0.121 (0.039)	0.116 (0.045)
Algorithme 6	0.148 (0.045)	0.149 (0.036)	0.135 (0.048)
Modèle IRT	0.108 (0.033)	0.106 (0.040)	0.090 (0.049)
Modèle valeurs attendues	0.161 (0.051)	0.171 (0.038)	0.130 (0.055)

Tableau 4.22 Tableau comparatif des résultats de taux d'erreurs pour les données de Polytechnique (modèle compensatoire)

Modèles	obs.size=15 (σ)	obs.size=30 (σ)	obs.size=45 (σ)
Algorithme 1	0.320 (0.014)	0.302 (0.014)	0.301 (0.029)
Algorithme 2	0.314 (0.015)	0.303 (0.015)	0.302 (0.028)
Algorithme 3	0.318 (0.016)	0.307 (0.013)	0.322 (0.027)
Algorithme 4	0.330 (0.014)	0.327 (0.014)	0.334 (0.027)
Algorithme 5	0.296 (0.013)	0.291 (0.017)	0.295 (0.028)
Algorithme 6	0.312 (0.020)	0.303 (0.016)	0.303 (0.029)
Modèle IRT	0.298 (0.015)	0.292 (0.018)	0.290 (0.027)
Modèle valeurs attendues	0.314 (0.021)	0.309 (0.020)	0.304 (0.030)

meilleurs résultats avec tous les jeux de données. D'un autre côté, nous remarquons que l'algorithme 5, qui est une combinaison entre les données générées par l'IRT et les données d'entraînement, cet algorithme a donné des résultats près de ceux d'IRT.

4.5.2 Analyses pour les résultats avec le modèle conjonctif

Nous présentons dans cette partie les résultats obtenus pour le modèle conjonctif. Nous comparons les résultats pour chaque jeu de données.

Le tableau 4.23 présente les résultats des taux d'erreurs obtenus avec les jeux de données de Vomlel. Nous avons pris différentes tailles de données d'observation (obs.size) soient 05, 10 et 15. Nous remarquons que seul l'algorithme 6 a donné les meilleurs résultats avec obs.size = 05. Par contre, les autres algorithmes ont donné des meilleurs résultats avec obs.size=15. En ce qui concerne obs.size = 10, nous remarquons qu'il n'y a pas un algorithme qui a bien performé.

Le tableau 4.24 présente les résultats des taux d'erreur obtenus avec le jeu de données

Tableau 4.23 Tableau comparatif des résultats de taux d'erreurs pour les données de Vomlel (modèle conjonctif)

Modèles	obs.size=05 (σ)	obs.size=10 (σ)	obs.size=15 (σ)
Algorithme 1	0.235 (0.036)	0.207 (0.086)	0.196 (0.034)
Algorithme 2	0.242 (0.033)	0.206 (0.087)	0.195 (0.040)
Algorithme 3	0.238 (0.032)	0.208 (0.097)	0.198 (0.038)
Algorithme 4	0.234 (0.037)	0.207 (0.096)	0.198 (0.037)
Algorithme 5	0.151 (0.022)	0.164 (0.082)	0.161 (0.040)
Algorithme 6	0.239 (0.037)	0.212 (0.089)	0.193 (0.037)
Modèle IRT	0.144 (0.023)	0.154 (0.083)	0.130 (0.030)
Modèle valeurs attendues	0.247 (0.040)	0.245 (0.085)	0.223 (0.055)

d'Unix. Nous avons pris différentes valeurs de tailles d'observation (obs.size) soient 10, 20 et 25. Nous remarquons que les algorithmes 3 et 4 ont donné des meilleurs résultats avec obs.size = 10. Par contre, ils ont donné, encore une fois, les mauvais résultats avec obs.size = 20 et surtout obs.size = 25. Le reste des algorithmes ont donné des meilleurs résultats avec obs.size=20. En ce qui concerne obs.size = 25, nous remarquons qu'il n'y a pas un algorithme qui a bien performé.

Revenons sur le modèle compensatoire, nous pouvons dire, que les algorithmes 3 et 4 ont donné toujours les meilleurs résultats avec le choix de obs.size=05, par contre tous les autres algorithmes qui ont bien performé avec obs.size=25 pour le modèle compensatoire, ils ont donné les meilleurs résultats avec obs.size=20 pour modèle conjonctif.

Le tableau 4.25 présente les résultats des taux d'erreur obtenus avec le jeu de données de Polytechnique. Nous avons pris différentes tailles de données d'observation (obs.size) soient 15, 30 et 45. Nous remarquons que l'algorithme 1 a donné les meilleurs résultats avec obs.size = 15. Pour les algorithmes 1, 2, 3, 5, 6 et modèle des valeurs attendues, ils ont donné des meilleurs résultats avec obs.size = 30. Enfin, seul le modèle IRT a donné de bons résultats avec obs.size=45. L'algorithme 4 est toujours le moins stable. Les algorithmes 1, 2 et 5 semblent plus stables et l'algorithme 5 et encore une autre fois celui qui donne les meilleurs résultats par rapport aux algorithmes 1, 2, 3, 4 et 6. Le IRT demeure toujours le meilleur avec n'importe quel choix de obs.size.

Nous pouvons confirmer que pour tous les algorithmes, l'IRT est celui qui a donné les meilleurs résultats avec tous les jeux de données et pour tout choix de obs.size. Nous remarquons aussi que l'algorithme 5 a donné des résultats presque proches de ceux d'IRT. Pour les autres algorithmes, les algorithmes 1 et 2 semblent les plus stables, l'algorithme 6 a donné de bons résultats nettement mieux que ceux du modèle des valeurs attendues, mais avec le jeu de données d'Unix, il semble moins stable.

Tableau 4.24 Tableau comparatif des résultats de taux d'erreurs pour les données d'Unix (modèle conjonctif)

Modèles	obs.size=10 (σ)	obs.size=20 (σ)	obs.size=25 (σ)
Algorithme 1	0.149 (0.024)	0.129 (0.039)	0.150 (0.046)
Algorithme 2	0.150 (0.029)	0.130 (0.037)	0.158 (0.044)
Algorithme 3	0.156 (0.036)	0.194 (0.057)	0.273 (0.095)
Algorithme 4	0.166 (0.038)	0.241 (0.077)	0.325 (0.114)
Algorithme 5	0.125 (0.026)	0.114 (0.026)	0.129 (0.042)
Algorithme 6	0.151 (0.032)	0.122 (0.039)	0.159 (0.047)
Modèle IRT	0.118 (0.033)	0.090 (0.033)	0.110 (0.043)
Modèle valeurs attendues	0.155 (0.043)	0.119 (0.046)	0.175 (0.046)

Tableau 4.25 Tableau comparatif des résultats de taux d'erreurs pour les données de Polytechnique (modèle conjonctif)

Modèles	obs.size=15 (σ)	obs.size=30 (σ)	obs.size=45 (σ)
Algorithme 1	0.322 (0.017)	0.306 (0.023)	0.307 (0.030)
Algorithme 2	0.318 (0.017)	0.305 (0.021)	0.307 (0.029)
Algorithme 3	0.320 (0.017)	0.310 (0.021)	0.323 (0.029)
Algorithme 4	0.325 (0.014)	0.326 (0.021)	0.343 (0.031)
Algorithme 5	0.296 (0.015)	0.294 (0.019)	0.300 (0.031)
Algorithme 6	0.318 (0.016)	0.307 (0.022)	0.313 (0.032)
Modèle IRT	0.298 (0.014)	0.291 (0.018)	0.289 (0.032)
Modèle valeurs attendues	0.314 (0.019)	0.307 (0.021)	0.310 (0.035)

4.5.3 Analyses de comparaison de modèle compensatoire et modèle conjonctif

Dans cette section, nous comparons les résultats obtenus pour les deux modèles.

Le tableau 4.26 présente les résultats des taux d'erreur obtenus avec les données de Vomlel en prenant obs.size=05. Nous remarquons que les algorithmes ont bien performé avec le modèle compensatoire. *Il est à noter que le modèle IRT n'est pas concerné par le mode conjonctif/compensatoire, c'est un modèle à part entier, on a également obtenu presque les mêmes résultats dans les deux cas.* Les résultats obtenus avec le modèle conjonctif sont nettement mieux que ceux avec le modèle compensatoire. Comme nous avons déjà expliqué, ce choix de obs.size = 05 ne donne pas vraiment des résultats précis surtout que obs.size est égal au nombre de facteurs latents.

Le tableau 4.27 présente les résultats des taux d'erreur obtenus avec les données de Vomlel en prenant obs.size=10. Nous remarquons que les résultats obtenus avec le modèle compensatoire sont nettement mieux que ceux avec le modèle conjonctif. Le IRT demeure toujours le meilleur par rapport aux modèles conjonctif et compensatoire.

Tableau 4.26 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Vomlel avec ops.size=5

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	5	0.228 (0.037)	0.235 (0.036)
Algorithme 2	5	0.227 (0.038)	0.242 (0.033)
Algorithme 3	5	0.227 (0.036)	0.238 (0.032)
Algorithme 4	5	0.228 (0.037)	0.234 (0.037)
Algorithme 5	5	0.158 (0.026)	0.151 (0.022)
Algorithme 6	5	0.235 (0.039)	0.239 (0.037)
Modèle IRT	5	0.155 (0.023)	0.144 (0.023)
Modèle valeurs attendues	5	0.248 (0.042)	0.247 (0.040)

Tableau 4.27 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Vomlel avec ops.size=10

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	10	0.186 (0.079)	0.207 (0.086)
Algorithme 2	10	0.189 (0.082)	0.206 (0.087)
Algorithme 3	10	0.186 (0.084)	0.208 (0.097)
Algorithme 4	10	0.185 (0.085)	0.207 (0.096)
Algorithme 5	10	0.152 (0.079)	0.164 (0.082)
Algorithme 6	10	0.192 (0.082)	0.212 (0.089)
Modèle IRT	10	0.151 (0.079)	0.154 (0.083)
Modèle valeurs attendues	10	0.221 (0.080)	0.245 (0.085)

Le tableau 4.28 présente les résultats des taux d'erreur obtenus avec les données de Vomlel en prenant obs.size=15. Nous remarquons qu'à l'exception du modèle des valeurs attendues qui a donné les mauvais résultats par rapport aux autres algorithmes, tous les algorithmes 1, 2, 3, 4, 5 et 6 ont bien performé avec les deux modèles compensatoire et conjonctif. Enfin, le IRT donne toujours les meilleurs résultats.

Le tableau 4.29 présente les résultats des taux d'erreur obtenus avec les données d'Unix en prenant obs.size=10. Nous remarquons que les résultats obtenus sont proches pour les deux modèles compensatoire et conjonctif.

Le tableau 4.30 contient les résultats des taux d'erreur obtenus avec les données d'Unix en prenant obs.size=20. Étrangement, nous remarquons que le modèle conjonctif a donné de bons résultats pour tous les algorithmes. Les résultats obtenus peuvent être expliqués par la petite taille du jeu de données, mais cette explication demeure spéculative. Enfin, comme d'habitude le modèle IRT reste toujours le meilleur.

Le tableau 4.31 contient les résultats des taux d'erreur obtenus avec les données d'Unix en

Tableau 4.28 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Vomlel avec ops.size=15

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	15	0.190 (0.046)	0.196 (0.034)
Algorithme 2	15	0.189 (0.047)	0.195 (0.040)
Algorithme 3	15	0.183 (0.054)	0.198 (0.038)
Algorithme 4	15	0.182 (0.055)	0.198 (0.037)
Algorithme 5	15	0.148 (0.038)	0.161 (0.040)
Algorithme 6	15	0.190 (0.044)	0.193 (0.037)
Modèle IRT	15	0.133 (0.054)	0.130 (0.030)
Modèle valeurs attendues	15	0.228 (0.047)	0.223 (0.055)

Tableau 4.29 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données d'Unix avec ops.size=10

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	10	0.155 (0.031)	0.149 (0.024)
Algorithme 2	10	0.155 (0.034)	0.150 (0.029)
Algorithme 3	10	0.155 (0.036)	0.156 (0.036)
Algorithme 4	10	0.164 (0.038)	0.166 (0.038)
Algorithme 5	10	0.116 (0.031)	0.125 (0.026)
Algorithme 6	10	0.148 (0.045)	0.151 (0.032)
Modèle IRT	10	0.108 (0.033)	0.118 (0.033)
Modèle valeurs attendues	10	0.161 (0.051)	0.155 (0.043)

Tableau 4.30 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données d'Unix avec ops.size=20

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	20	0.146 (0.042)	0.129 (0.039)
Algorithme 2	20	0.142 (0.043)	0.130 (0.037)
Algorithme 3	20	0.212 (0.086)	0.194 (0.057)
Algorithme 4	20	0.265 (0.098)	0.241 (0.077)
Algorithme 5	20	0.121 (0.039)	0.114 (0.026)
Algorithme 6	20	0.149 (0.036)	0.122 (0.039)
Modèle IRT	20	0.106 (0.040)	0.090 (0.033)
Modèle valeurs attendues	20	0.171 (0.038)	0.119 (0.046)

prenant obs.size=25. Nous constatons que le modèle conjonctif donne des résultats proches de ceux de modèle compensatoire pour certains algorithmes. Le modèle IRT et l’algorithme 5 sont toujours les meilleurs.

Le tableau 4.32 contient les résultats des taux d’erreur obtenus avec le jeu de données de Polytechnique en prenant obs.size=15. Nous remarquons que sauf pour l’algorithme 4 et le modèle des valeurs attendues, tous les algorithmes ont bien performé avec le modèle compensatoire.

Le tableau 4.33 contient les résultats des taux d’erreur obtenus avec le jeu de données de Polytechnique en prenant obs.size=30. Nous remarquons que les algorithmes ont donné de bons résultats aussi bien avec le modèle compensatoire qu’avec le modèle conjonctif.

Le tableau 4.34 contient les résultats des taux d’erreur obtenus avec le jeu de données de Polytechnique en prenant obs.size=45. Nous constatons que tous les algorithmes ont donné de bons résultats avec les deux modèles compensatoire et conjonctif. Le IRT demeure toujours le meilleur.

4.5.4 Analyses générales

Dans cette section nous présentons une interprétation générale à partir des analyses précédemment établies.

- Tous les modèles donnent en général des résultats meilleurs que le modèle de base des valeurs attendues. Ceci démontre la valeur de ces modèles.
- Cependant, il est clair que le modèle IRT donne toujours les meilleurs résultats par rapport aux deux modèles conjonctif ou compensatoire. Malgré qu’il ne traite qu’une seule compétence à la fois, ceci démontre que la capacité de représenter la difficulté et la discrimination des questions avec une régression logistique est plus performante que les modèles linéaires étudiés.

Tableau 4.31 Tableau comparatif des résultats de taux d’erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données d’Unix avec ops.size=25

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	25	0.132 (0.043)	0.150 (0.046)
Algorithme 2	25	0.132 (0.048)	0.158 (0.044)
Algorithme 3	25	0.276 (0.089)	0.273 (0.095)
Algorithme 4	25	0.323 (0.094)	0.325 (0.114)
Algorithme 5	25	0.116 (0.045)	0.129 (0.042)
Algorithme 6	25	0.135 (0.048)	0.159 (0.047)
Modèle IRT	25	0.090 (0.049)	0.110 (0.043)
Modèle valeurs attendues	25	0.130 (0.055)	0.175 (0.046)

Tableau 4.32 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Polytechnique avec ops.size=15

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	15	0.320 (0.014)	0.322 (0.017)
Algorithme 2	15	0.314 (0.015)	0.318 (0.017)
Algorithme 3	15	0.318 (0.016)	0.320 (0.017)
Algorithme 4	15	0.330 (0.014)	0.325 (0.014)
Algorithme 5	15	0.296 (0.013)	0.296 (0.015)
Algorithme 6	15	0.312 (0.020)	0.318 (0.016)
Modèle IRT	15	0.298 (0.015)	0.298 (0.014)
Modèle valeurs attendues	15	0.314 (0.021)	0.307 (0.019)

Tableau 4.33 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Polytechnique avec ops.size=30

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	30	0.302 (0.014)	0.306 (0.023)
Algorithme 2	30	0.303 (0.015)	0.305 (0.021)
Algorithme 3	30	0.307 (0.013)	0.310 (0.021)
Algorithme 4	30	0.327 (0.014)	0.326 (0.021)
Algorithme 5	30	0.291 (0.017)	0.294 (0.019)
Algorithme 6	30	0.303 (0.016)	0.307 (0.022)
Modèle IRT	30	0.292 (0.018)	0.291 (0.018)
Modèle valeurs attendues	30	0.309 (0.020)	0.307 (0.021)

Tableau 4.34 Tableau comparatif des résultats de taux d'erreurs entre le modèle conjonctif et le modèle compensatoire en utilisant les données de Polytechnique avec ops.size=45

Modèles	obs.size	Modèle compensatoire	Modèle conjonctif
Algorithme 1	45	0.301 (0.029)	0.307 (0.030)
Algorithme 2	45	0.302 (0.028)	0.307 (0.029)
Algorithme 3	45	0.322 (0.027)	0.323 (0.029)
Algorithme 4	45	0.334 (0.027)	0.343 (0.031)
Algorithme 5	45	0.295 (0.028)	0.300 (0.031)
Algorithme 6	45	0.303 (0.029)	0.313 (0.032)
Modèle IRT	45	0.290 (0.027)	0.289 (0.032)
Modèle valeurs attendues	45	0.304 (0.030)	0.310 (0.035)

- Avec l’algorithme 5, qui est un modèle hybride, nous avons trouvé des résultats proches de ceux de IRT.
- Les deux algorithmes 1 et 2 semblent plus stables par rapport aux algorithmes 3 et 4.
- L’algorithme 4 est le moins stable
- Les deux modèles conjonctifs et compensatoires donnent des résultats similaires. Par contre, on s’attendrait que le modèle conjonctif donne des résultats moins bons par rapport à celui de compensatoire. Puisque dans le premier modèle la condition pour qu’un étudiant réussisse la question il doit maîtriser les deux compétences en même temps.
- Il semble que le modèle conjonctif et avec les petits jeux comportant peu de données donne de bons résultats, mais sous certaines conditions et avec le bon choix de obs.size.

CHAPITRE 5

CONCLUSION

Dans cette section, nous ferons une synthèse du travail effectué. Nous présentons également les limites de notre recherche et mettons l'accent sur d'autres pistes d'améliorations pour de futurs travaux de recherche.

5.1 Synthèse du déroulement du projet

Afin de réaliser notre projet, nous avons commencé par un survol du domaine de l'apprentissage assisté par ordinateur et l'évaluation automatique des étudiants. Notre projet porte sur un guide d'étude qui permet aux étudiants d'évaluer leurs compétences. La partie de revue de littérature nous a permis de comprendre le principe de diagnostic des compétences des étudiants et surtout l'importance de l'alignement des questions avec les compétences impliquées afin de réaliser un diagnostic précis et efficace. Nous avons étudié également les différents modèles d'alignement et les différentes techniques et méthodes statistiques qui permettent de générer cet alignement automatiquement comme la méthode de factorisation de matrice en valeurs non négatives ou le NMF.

Puisque le but de notre projet est de fournir un diagnostic détaillé à l'étudiant quant à ses compétences et son niveau de préparation, nous avons étudié le principe d'alignement, pour ce faire, nous avons choisi de nous focaliser sur deux modèles d'alignement qui sont le modèle conjonctif et le modèle compensatoire. Et afin d'étudier la validité de prédiction de ces deux modèles nous avons proposé quelques algorithmes. L'étude des autres modèles comme le modèle de théorie de réponses aux items IRT et le modèle de valeurs attendues, nous a permis également de proposer des algorithmes hybrides, qui se basent sur la combinaison de données d'entraînement avec les jeux de données générées par le modèle IRT ou par le modèle des valeurs attendues.

Pour valider les algorithmes définis, nous avons effectué des simulations afin de valider nos algorithmes et surtout d'évaluer la validité de prédiction de nos modèles. Nous avons aussi effectué les comparaisons avec les modèles IRT et le modèle de valeurs attendues.

Les résultats obtenus montrent que le modèle IRT reste toujours celui qui donne les meilleurs résultats. Par contre, avec l'algorithme 5, nous avons pu générer des résultats proches à ceux du IRT dans les deux modèles conjonctif et compensatoire. Ces résultats sont importants et nous a permis d'ouvrir une autre porte de recherche qui peut être intéressante.

En ce qui concerne les deux modèles, le modèle compensatoire a mieux performé pour deux jeux de données sur trois.

En général, les résultats obtenus pour les deux modèles et pour la plupart des algorithmes sont proches. Par contre, certains algorithmes semblent instables comme l’algorithme 4 et l’algorithme 3, surtout avec le jeu de données d’Unix.

5.2 Perspectives et travaux futurs

Malgré que nous n’ayons pas réussi à fournir un modèle plus performant que le IRT, les résultats obtenus ne sont pas néanmoins sans intérêt.

D’une part, les modèles linéaires permettent de décomposer la performance en un ensemble de compétences distinctes, alors que le modèle IRT se limite à une seule compétence. Pour un guide d’étude, par exemple, cette décomposition est importante puisqu’elle permet d’identifier les sujets non maîtrisés. Il est plus utile de spécifier que tel ou tel autre sujet est moins bien maîtrisé, que de simplement dire que le niveau global est, ou n’est pas atteint.

D’autre part, les résultats démontrent qu’en combinant la méthode IRT avec un modèle linéaire, la précision des prédictions, est alors près de celle de cette méthode, tout en permettant une décomposition en compétences plus précise.

Pour finir, nous travaillons toujours sur le guide d’étude, nous essayons de trouver les meilleurs modèles et solutions afin de pouvoir établir le diagnostic le plus précis que possible qui aidera l’étudiant à s’améliorer.

RÉFÉRENCES

- BIRNBAUM, A. (1968). Some latent trait models and their use in inferring an examinee's ability, 397–472.
- CONATI, C., GERTNER, A. et VANLEHN, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12, 371–417.
- CONEJO, R., GUZMAN, E., MILLÁN, E., TRELLA, M., PÉREZ-DE-LA CRUZ, J. L. et RIOS, A. (2004). SIETTE : A web-based tool for adaptive teaching. *International Journal of Artificial Intelligence in Education*, 14, 29–61.
- CORBETT, A. T. et ANDERSON, J. R. (1995). Knowledge Tracing : Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, vol. 4, pp. 253–278.
- DESMARAIS, M., PU, X. et BLAIS, J.-G. (2007). Partial order knowledge structure for cat applications. D. Weiss, éditeur, *Computerized Adaptive Testing, GMAC 2008 Proceedings*. (18 pages, on line).
- DESMARAIS, M. C. et BAKER, R. (2011). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interactions*, 22, 9–38.
- DESMARAIS, M. C., BEHESHTI, B. et NACEUR, R. (2012). Item to skills mapping : Deriving a conjunctive Q-matrix from data. *11th Conference on Intelligent Tutoring Systems, ITS 2012*. pp. 454–463.
- DESMARAIS, M. C. et GAGNON, M. (2006). Bayesian student models based on item to item knowledge structures. W. Nejdl et K. Tochtermann, éditeurs, *LNCSE : Innovative Approaches for Learning and Knowledge Sharing (EC-TEL 2006)*. 111–124.
- DESMARAIS, M. C., GAGNON, M. et MESHKINFAM, P. (2006). Item to item student models. *AAAI'2006 Workshop 06, Data mining in education*. pp. 10–20.
- DOIGNON, J.-P. et FALMAGNE, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, 23, 175–196.
- FALMAGNE, J.-C., COSYN, E., DOIGNON, J.-P. et THIÉRY, N. (2006). The assessment of knowledge, in theory and in practice. R. Missaoui et J. Schmid, éditeurs, *ICFCA*. Springer, vol. 3874 de *Lecture Notes in Computer Science*, 61–79.
- HOCKEMEYER, C., HELD, T. et ALBERT, D. (1997). RATH - a relational adaptive tutoring hypertext WWW-environment based on knowledge space theory.

KOEDINGER, K. R., ANDERSON, J. R., HADLEY, W. H. et MARK, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30–43.

MISLEVY, R. J. et GITOMER, D. (1995). The role of probability-based inference in an intelligent tutoring system. *User Modeling and User-Adapted Interaction*, 42, 253–282.

MITROVIC, A. (2012). Fifteen years of Constraint-Based Tutors : What we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, vol. 21, pp. 39–72.

MITROVIC, A., KOEDINGER, K. R. et MARTIN, B. (2003). A comparative analysis of cognitive tutoring and constraint-based modeling. *User Modeling 2003, 9th International Conference, UM 2003, Johnstown, PA, USA, June 22-26, 2003*. Springer, 313–322.

RIZOPOULOS, D. (2006). ltm : An R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, 17, 1–25.

VANLEHN, K., LYNCH, C., SCHULZE, K., SHAPIRO, J. A., SHELBY, R., TAYLOR, L., TREACY, D., WEINSTEIN, A. et WINTERSGILL, M. (2005). The andes physics tutoring system : Five years of evaluation. *Proceedings of AIED'05*. 678–685.

VOMLEL, J. (2004). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 12, 83–100.

ANNEXE A

Exemple du résultat d'un test

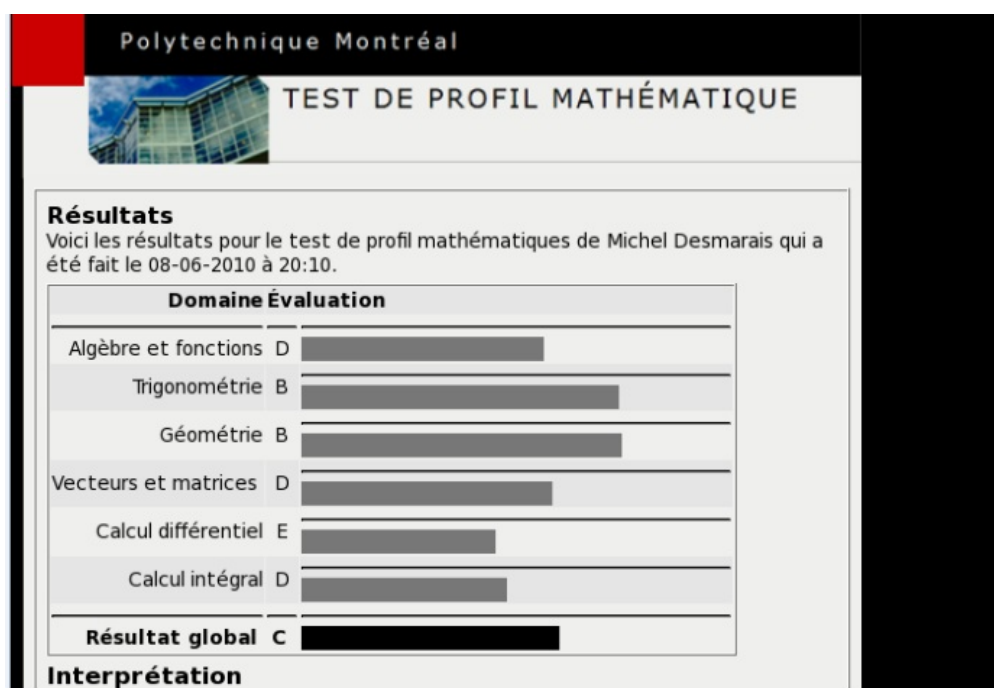


Figure A.1 Résultats du test fait par l'étudiant


Résultat global C	
--------------------------	--

Interprétation

- Votre note globale est A+ ou A
 - Vous semblez posséder les outils mathématiques nécessaires à la réussite de vos études à l'École Polytechnique.
 - Passez de bonnes vacances!
- **Votre note globale est B ou C**
 - **Vos chances de réussite à l'École Polytechnique sont bonnes.**
 - **Il faudrait cependant réviser certaines notions et techniques mathématiques dans les domaines où vos résultats sont les moins bons dans votre profil affiché ci-dessus.**
- Votre note globale est D ou E
 - Nos statistiques démontrent que vous pourriez être dans un groupe à risque d'échec dans certains cours obligatoires dans les programmes de l'École Polytechnique.
 - Nous vous suggérons donc de réviser l'ensemble de vos connaissances mathématiques pour les mettre à jour.
 - Nous vous recommandons de suivre le [cours de révision Z-050 Mathématiques fondamentales](#) qui aura lieu du mercredi 21 juillet au vendredi 13 août 2010, c'est-à-dire en mode intensif. [Pour vous inscrire, complétez le formulaire téléchargeable par ce lien.](#)
 - Nous vous recommandons de suivre le [cours de révision Z-050 Mathématiques fondamentales](#) qui aura lieu du mercredi 21 juillet au vendredi 13 août 2010, c'est-à-dire en mode intensif. [Pour vous inscrire, complétez le formulaire téléchargeable par ce lien.](#)
 - Nous vous proposons de réduire le nombre total de crédits des cours à suivre au cours de la session d'automne afin de faciliter votre apprentissage en mathématiques.

Élaborer une stratégie personnelle de réussite

Tous ceux qui désirent élaborer une stratégie personnelle de réussite ont la possibilité de discuter avec M. Philippe Razanakolona, conseiller aux étudiants de première année. Vous pouvez le rejoindre au Bureau du Soutien aux études en composant le 514-340-4711, poste 4205, ou en écrivant à : sep-reussite@polymtl.ca



ÉCOLE
POLYTECHNIQUE
MONTRÉAL

© École Polytechnique de Montréal, 2010

[Haut de page](#)

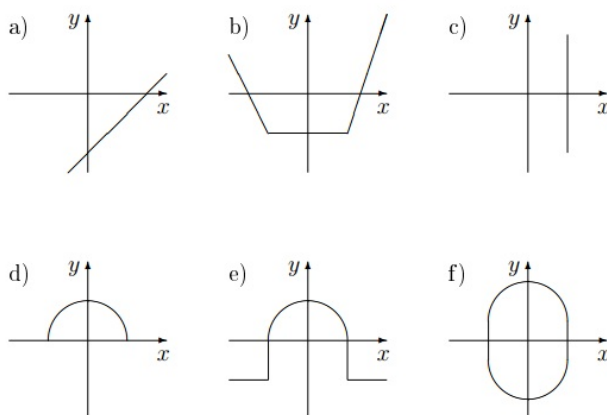
Figure A.2 Courriel transmis aux étudiants après le test

ANNEXE B

Exemple du questionnaire de polytechnique

Dans cette annexe on vous présente un exemple de questionnaire, qui est celui de polytechnique. Voici un exemple de questions telles que posées lors du questionnaire de mathématique :

1. Soit les graphes suivants

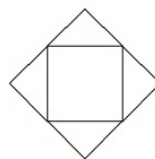


Les graphes représentant des fonctions sont :

1. Seulement (a), (b) et (d) Seulement (a), (b), (c) et (d)
2. Tous
3. Tous sauf (c) et (f)
4. Je ne sais pas

2. Si l'aire du carré intérieur est de 6 cm^2 , l'aire du carré extérieur est de

- 1) 9 cm^2
- 2) 12 cm^2
- 3) 24 cm^2
- 4) 36 cm^2
- 5) je ne sais pas



3. Soit la fonction

$$g(x) = \begin{cases} 1 & \text{si } x = 0 \\ \frac{1}{x} & \text{si } x \neq 0 \end{cases}.$$

Alors

- 1) $g(x)$ est continue partout
- 2) $g(x)$ est continue partout sauf en $x = 0$
- 3) $g(x)$ est continue partout sauf en $x = 1$
- 4) $g(x)$ est croissante pour $x > 0$
- 5) je ne sais pas

4. Soit $M(2, \pi/3)$ un point en coordonnées polaires. Soit x et y les coordonnées cartésiennes de M . Alors la distance de M à l'origine est

- 1) $x^2 + y^2$
- 2) 2
- 3) $2 \cos \pi/3$
- 4) 4
- 5) je ne sais pas

5. L'équation de la tangente au cercle d'équation $(x-1)^2 + (y-2)^2 = 4$ passant par le point $A(3, 2)$ est

- 1) $x = 2$
- 2) $y = 3$
- 3) $2x - 3y = 0$
- 4) autre chose
- 5) je ne sais pas

ANNEXE C

D'autres résultats

Résultats avec le modèle compensatoire

Résultats obtenus avec les données de Vomlel

Le tableau C.1 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.1.

Le tableau C.2 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.2.

Le tableau C.3 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.3.

Résultats obtenus avec les données d'Unix

Le tableau C.4 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.4.

Le tableau C.5 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.5.

Le tableau C.6 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.6.

Résultats obtenus avec les données de Polytechnique

Le tableau C.7 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.7.

Le tableau C.8 contient les valeurs numériques de taux d'erreur obtenu pour chaque

Tableau C.1 Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.1

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	128	5	0.215
Algorithme 2	3	24	128	5	0.218
Algorithme 3	3	24	128	5	0.217
Algorithme 4	3	24	128	5	0.214
Algorithme 5	3	24	128	5	0.155
Algorithme 6	3	24	128	5	0.218
Modèle IRT	3	24	128	5	0.148
Modèle valeurs attendues	3	24	128	5	0.228

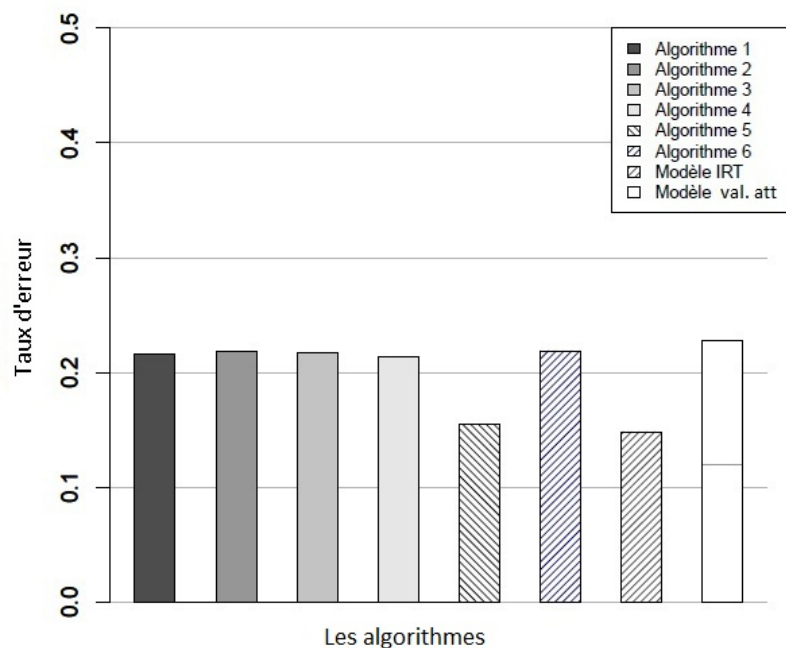


Figure C.1 Résultats de performance avec les données de Vomlel avec le modèle compensatoire, $n.skills = 3$ et $obs.size = 5$

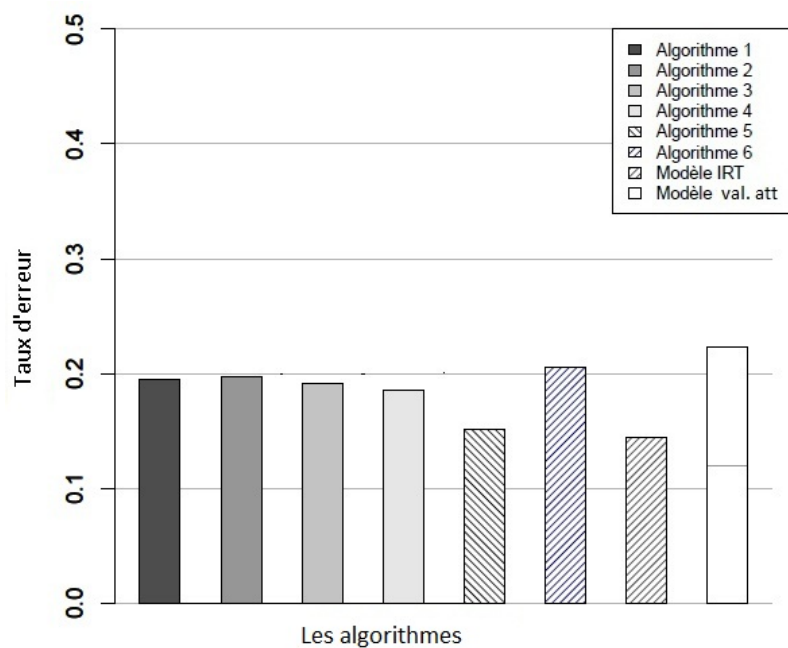


Figure C.2 Résultats de performance avec les données de Vomlel avec le modèle compensatoire, $n.skills = 3$ et $obs.size = 10$

Tableau C.2 Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.2

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	128	10	0.194
Algorithme 2	3	24	128	10	0.196
Algorithme 3	3	24	128	10	0.191
Algorithme 4	3	24	128	10	0.185
Algorithme 5	3	24	128	10	0.151
Algorithme 6	3	24	128	10	0.205
Modèle IRT	3	24	128	10	0.144
Modèle valeurs attendues	3	24	128	10	0.223

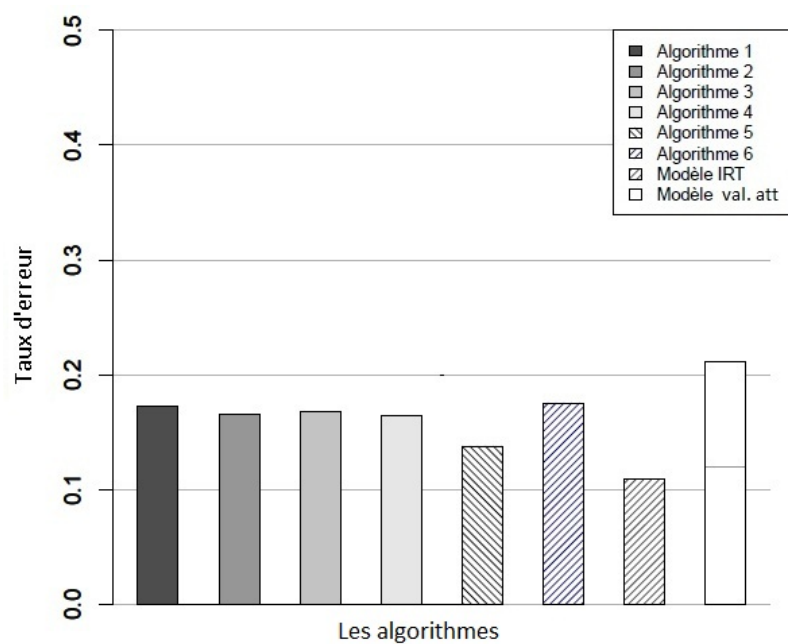


Figure C.3 Résultats de performance avec les données de Vomlel avec le modèle compensatoire, n.skills = 3 et obs.size = 15

Tableau C.3 Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.3

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	128	15	0.172
Algorithme 2	3	24	128	15	0.165
Algorithme 3	3	24	128	15	0.167
Algorithme 4	3	24	128	15	0.163
Algorithme 5	3	24	128	15	0.137
Algorithme 6	3	24	128	15	0.175
Modèle IRT	3	24	128	15	0.108
Modèle valeurs attendues	3	24	128	15	0.211

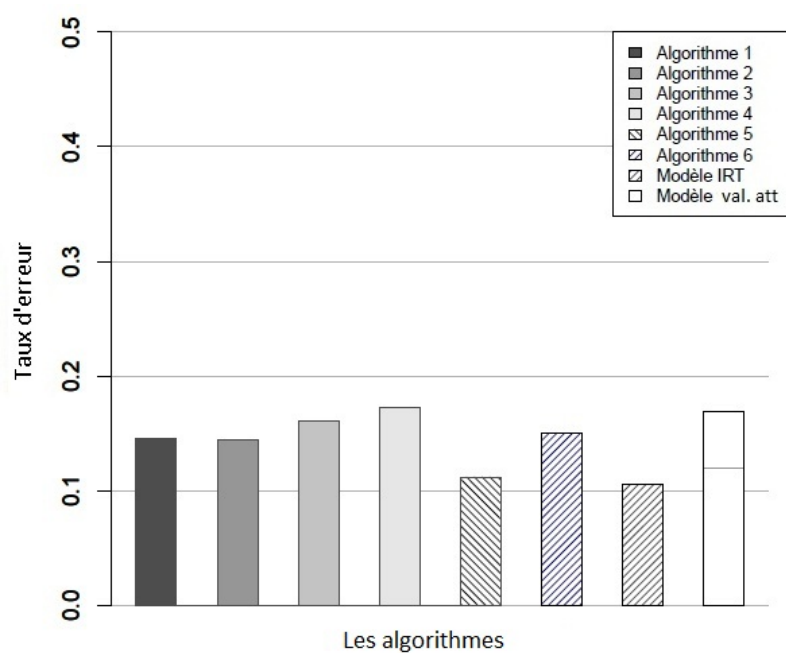


Figure C.4 Résultats de performance avec les données de Unix avec le modèle compensatoire, n.skills = 3 et obs.size = 10

Tableau C.4 Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.4

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	40	10	0.146
Algorithme 2	3	24	40	10	0.144
Algorithme 3	3	24	40	10	0.160
Algorithme 4	3	24	40	10	0.172
Algorithme 5	3	24	40	10	0.112
Algorithme 6	3	24	40	10	0.150
Modèle IRT	3	24	40	10	0.105
Modèle valeurs attendues	3	24	40	10	0.169

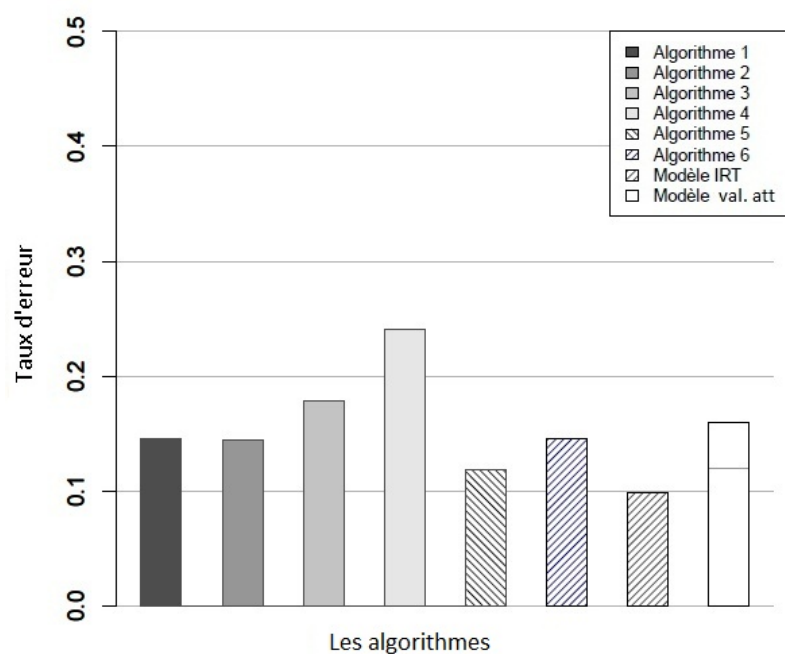


Figure C.5 Résultats de performance avec les données de Unix avec le modèle compensatoire, n.skills = 3 et obs.size = 20

Tableau C.5 Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.5

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	40	20	0.145
Algorithme 2	3	24	40	20	0.144
Algorithme 3	3	24	40	20	0.178
Algorithme 4	3	24	40	20	0.241
Algorithme 5	3	24	40	20	0.118
Algorithme 6	3	24	40	20	0.145
Modèle IRT	3	24	40	20	0.098
Modèle valeurs attendues	3	24	40	20	0.159

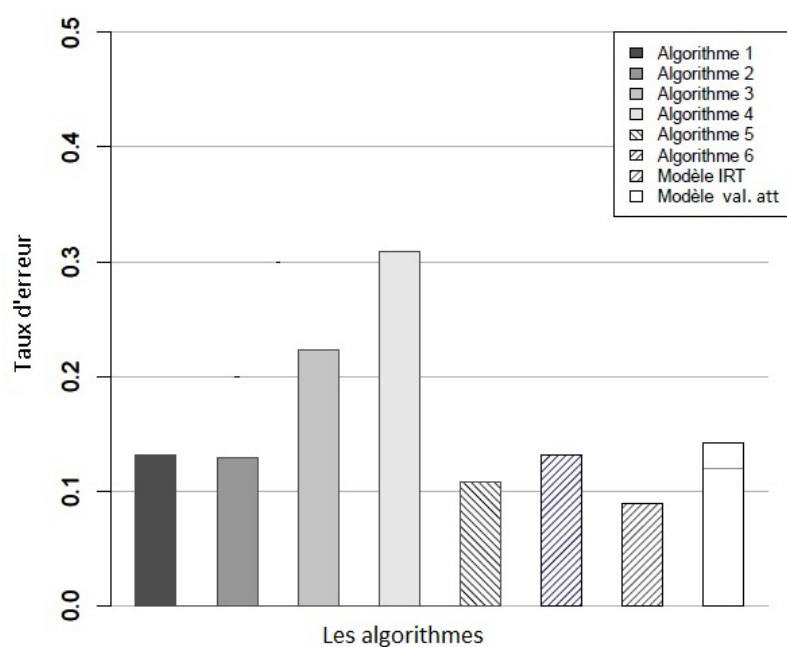


Figure C.6 Résultats de performance avec les données de Unix avec le modèle compensatoire, n.skills = 3 et obs.size = 25

Tableau C.6 Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.6

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	40	25	0.131
Algorithme 2	3	24	40	25	0.129
Algorithme 3	3	24	40	25	0.223
Algorithme 4	3	24	40	25	0.308
Algorithme 5	3	24	40	25	0.107
Algorithme 6	3	24	40	25	0.131
Modèle IRT	3	24	40	25	0.089
Modèle valeurs attendues	3	24	40	25	0.142

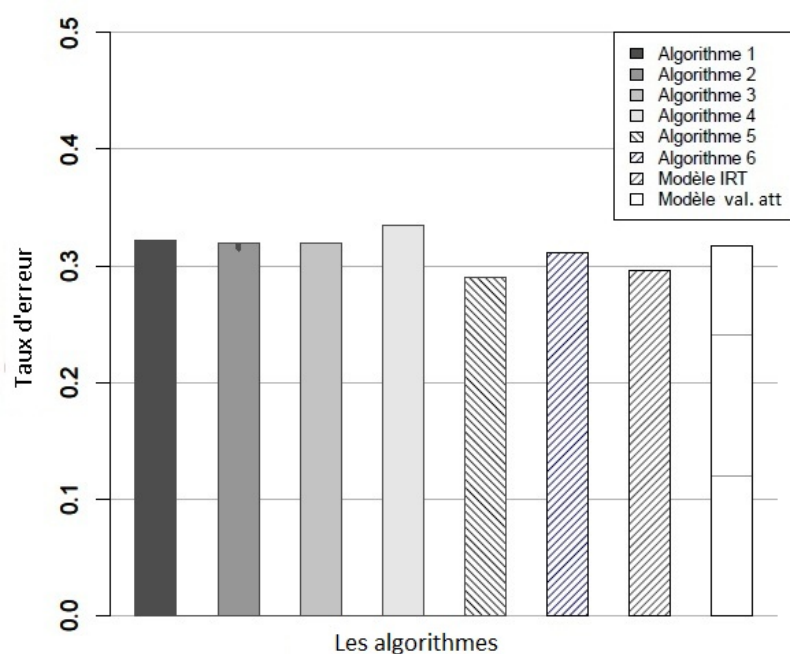


Figure C.7 Résultats de performance avec les données de Polytechnique avec le modèle compensatoire, n.skills = 3 et obs.size = 15

Tableau C.7 Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.7

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	200	15	0.321
Algorithme 2	3	24	200	15	0.319
Algorithme 3	3	24	200	15	0.334
Algorithme 4	3	24	200	15	0.334
Algorithme 5	3	24	200	15	0.289
Algorithme 6	3	24	200	15	0.310
Modèle IRT	3	24	200	15	0.295
Modèle valeurs attendues	5	24	200	15	0.317

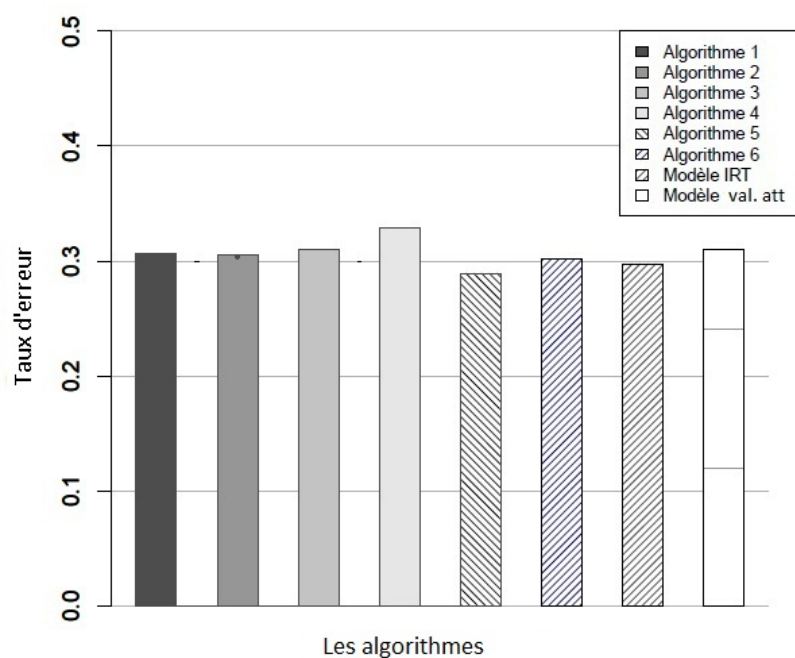


Figure C.8 Résultats de performance avec les données de Polytechnique avec le modèle compensatoire, n.skills = 3 et obs.size = 30

algorithme, relatif à la figure C.8.

Le tableau C.9 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.9.

Résultats avec le modèle conjonctif

Résultats obtenus avec les données de Vomlel

Le tableau C.10 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.10.

Le tableau C.11 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.11.

Le tableau C.12 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.12.

Résultats obtenus avec les données d'Unix

Le tableau C.13 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.13.

Le tableau C.14 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.14.

Le tableau C.15 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.15.

Résultats obtenus avec les données de Polytechnique

Le tableau C.16 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.16.

Le tableau C.17 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.17.

Le tableau C.18 contient les valeurs numériques de taux d'erreur obtenu pour chaque algorithme, relatif à la figure C.18.

Tableau C.8 Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.8

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	200	30	0.306
Algorithme 2	3	24	200	30	0.305
Algorithme 3	3	24	200	30	0.310
Algorithme 4	3	24	200	30	0.328
Algorithme 5	3	24	200	30	0.289
Algorithme 6	3	24	200	30	0.301
Modèle IRT	3	24	200	30	0.297
Modèle valeurs attendues	3	24	200	30	0.310

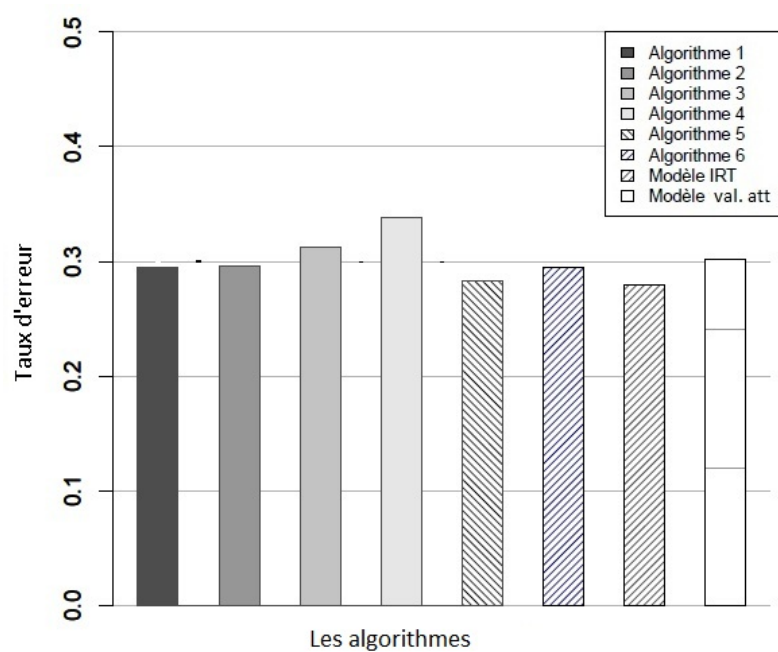


Figure C.9 Résultats de performance avec les données de Polytechnique avec le modèle compensatoire, n.skills = 3 et obs.size = 45

Tableau C.9 Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.9

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	200	45	0.294
Algorithme 2	3	24	200	45	0.296
Algorithme 3	3	24	200	45	0.312
Algorithme 4	3	24	200	45	0.337
Algorithme 5	3	24	200	45	0.282
Algorithme 6	3	24	200	45	0.294
Modèle IRT	3	24	200	45	0.279
Modèle valeurs attendues	3	24	200	45	0.302

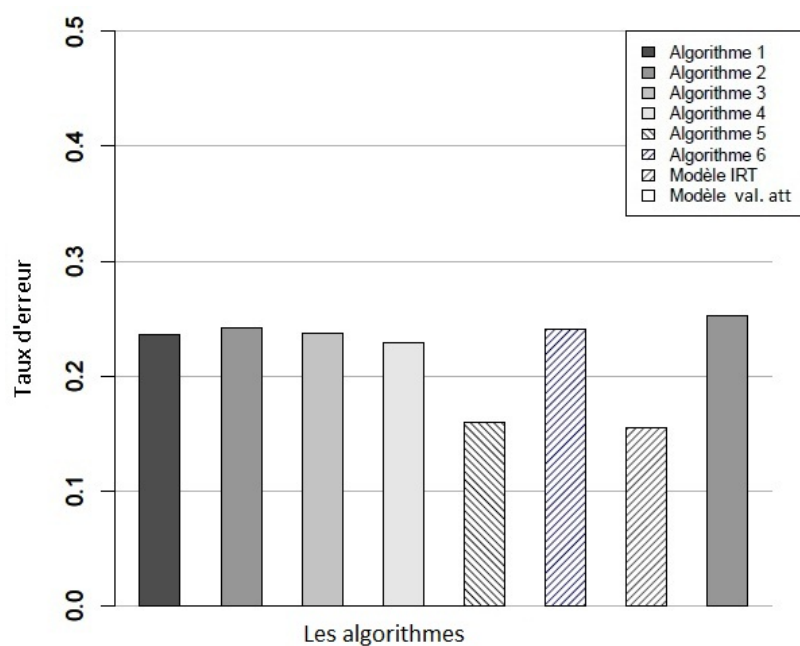


Figure C.10 Résultats de performance avec les données de Vomlel avec le modèle conjonctif, n.skills = 3 et obs.size = 5

Tableau C.10 Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.10

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	128	5	0.236
Algorithme 2	3	24	128	5	0.241
Algorithme 3	3	24	128	5	0.237
Algorithme 4	3	24	128	5	0.228
Algorithme 5	3	24	128	5	0.159
Algorithme 6	3	24	128	5	0.240
Modèle IRT	3	24	128	5	0.154
Modèle valeurs attendues	3	24	128	5	0.251

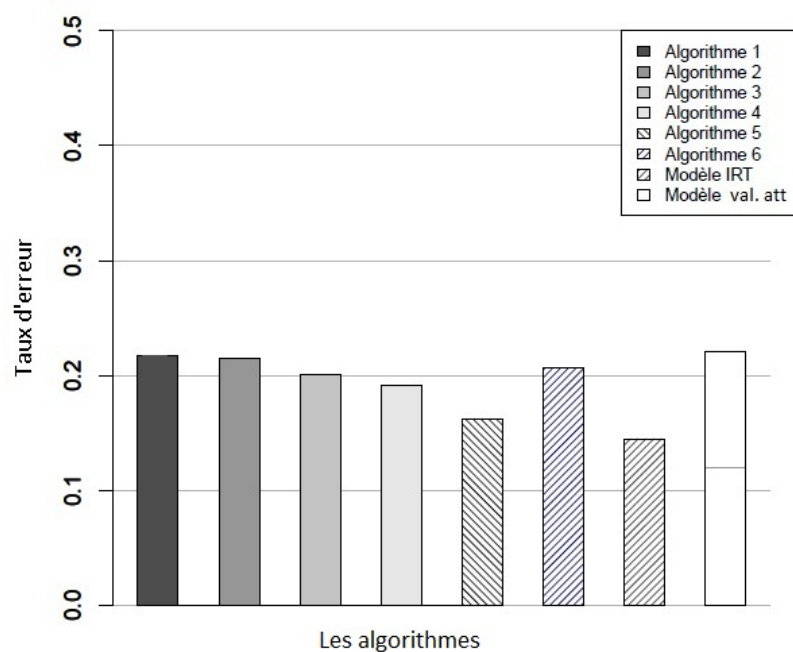


Figure C.11 Résultats de performance avec les données de Vomlel avec le modèle conjonctif, n.skills = 3 et obs.size = 10

Tableau C.11 Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.11

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	128	10	0.217
Algorithme 2	3	24	128	10	0.214
Algorithme 3	3	24	128	10	0.201
Algorithme 4	3	24	128	10	0.191
Algorithme 5	3	24	128	10	0.161
Algorithme 6	3	24	128	10	0.206
Modèle IRT	3	24	128	10	0.144
Modèle valeurs attendues	3	24	128	10	0.220

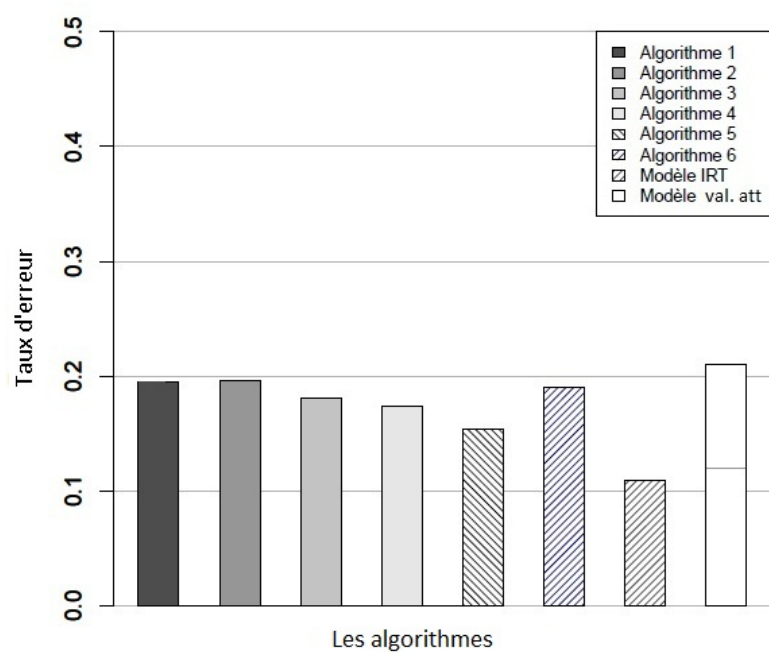


Figure C.12 Résultats de performance avec les données de Vomlel avec le modèle conjonctif, n.skills = 3 et obs.size = 15

Tableau C.12 Tableau de valeurs de prédiction avec les données de Vomlel, relatif à la figure C.12

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	128	15	0.195
Algorithme 2	3	24	128	15	0.195
Algorithme 3	3	24	128	15	0.181
Algorithme 4	3	24	128	15	0.174
Algorithme 5	3	24	128	15	0.153
Algorithme 6	3	24	128	15	0.190
Modèle IRT	3	24	128	15	0.109
Modèle valeurs attendues	3	24	128	15	0.209

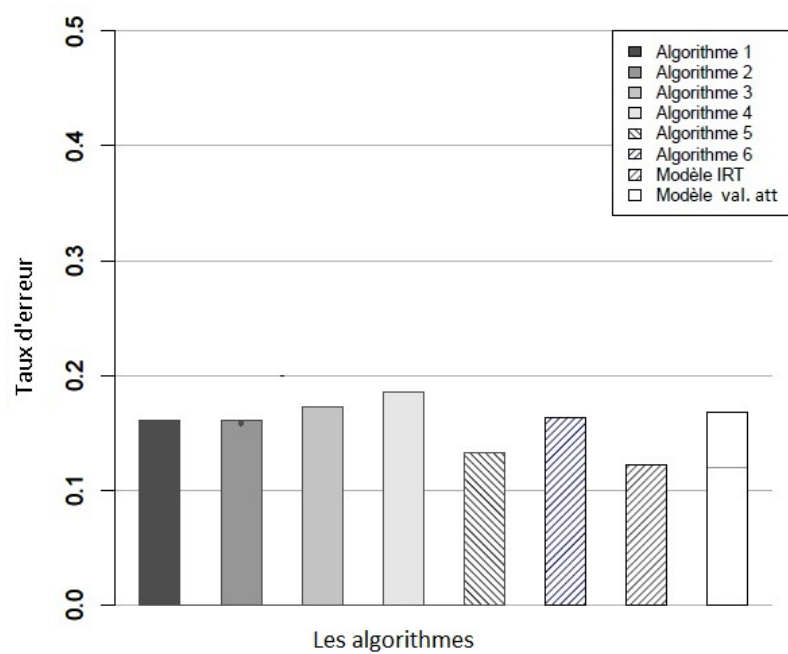


Figure C.13 Résultats de performance avec les données de Unix avec le modèle conjonctif, n.skills = 3 et obs.size = 10

Tableau C.13 Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.13

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	40	10	0.161
Algorithme 2	3	24	40	10	0.160
Algorithme 3	3	24	40	10	0.172
Algorithme 4	3	24	40	10	0.185
Algorithme 5	3	24	40	10	0.132
Algorithme 6	3	24	40	10	0.163
Modèle IRT	3	24	40	10	0.121
Modèle valeurs attendues	3	24	40	10	0.167

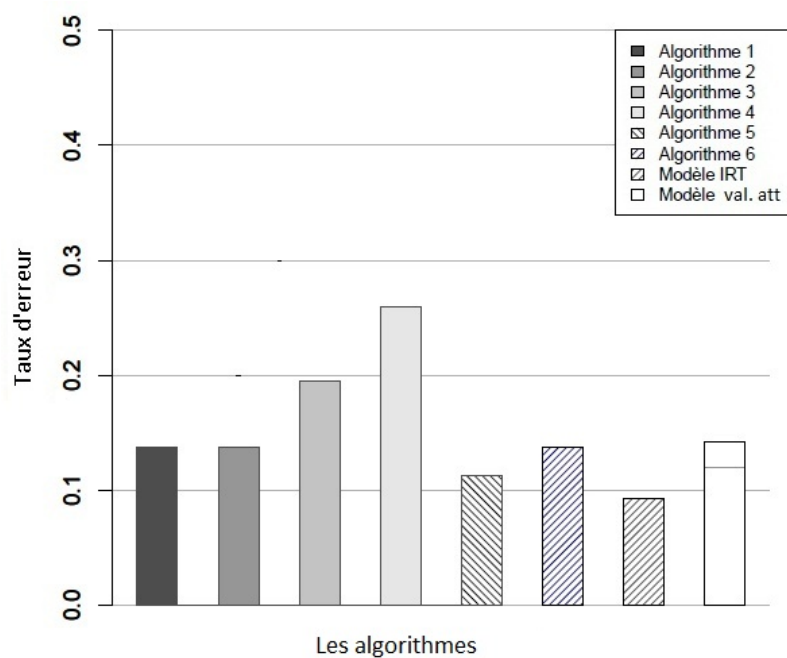


Figure C.14 Résultats de performance avec les données de Unix avec le modèle conjonctif, n.skills = 3 et obs.size = 20

Tableau C.14 Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.14

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	40	20	0.136
Algorithme 2	3	24	40	20	0.136
Algorithme 3	3	24	40	20	0.195
Algorithme 4	3	24	40	20	0.259
Algorithme 5	3	24	40	20	0.112
Algorithme 6	3	24	40	20	0.137
Modèle IRT	3	24	40	20	0.092
Modèle valeurs attendues	3	24	40	20	0.142

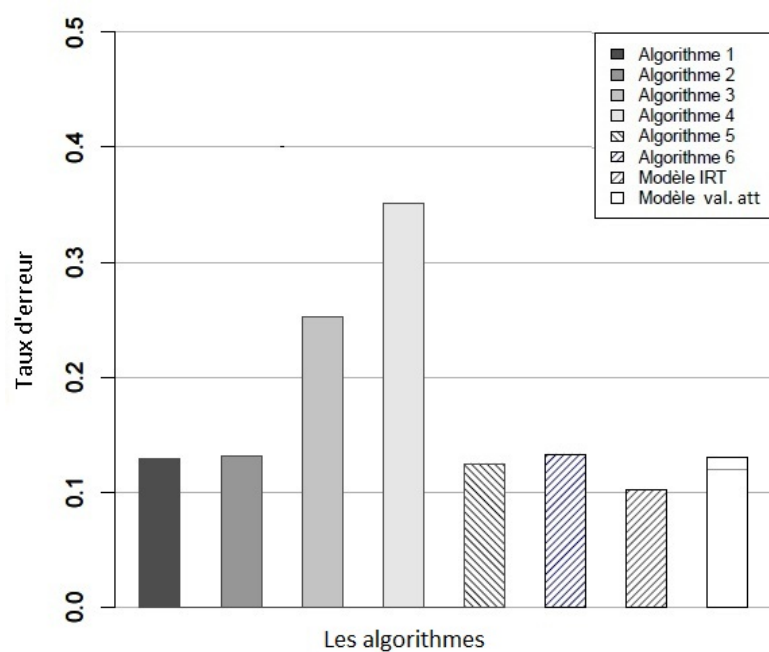


Figure C.15 Résultats de performance avec les données de Unix avec le modèle conjonctif, n.skills = 3 et obs.size = 25

Tableau C.15 Tableau de valeurs de prédiction avec les données d'Unix, relatif à la figure C.15

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	40	25	0.129
Algorithme 2	3	24	40	25	0.131
Algorithme 3	3	24	40	25	0.252
Algorithme 4	3	24	40	25	0.351
Algorithme 5	3	24	40	25	0.125
Algorithme 6	3	24	40	25	0.133
Modèle IRT	3	24	40	25	0.101
Modèle valeurs attendues	3	24	40	25	0.130

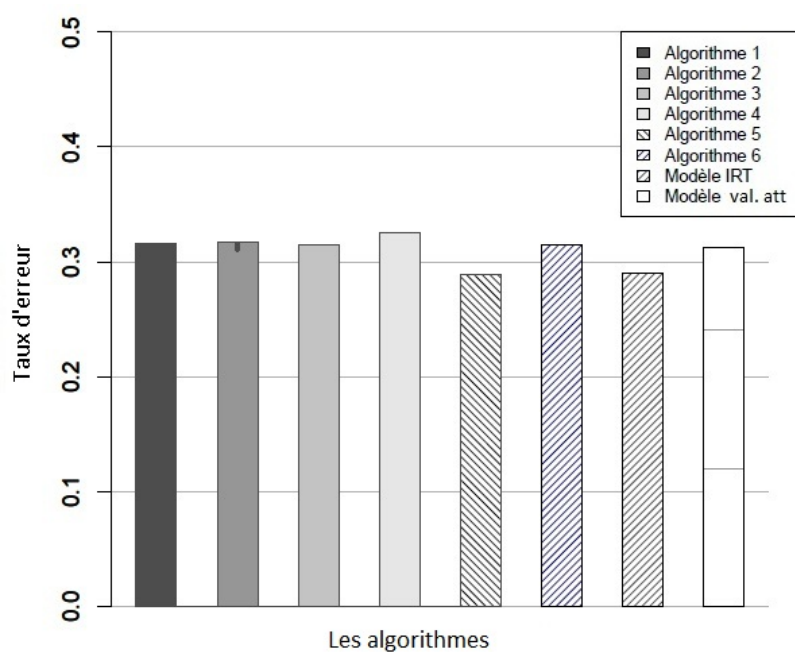


Figure C.16 Résultats de performance avec les données de Polytechnique avec le modèle conjonctif, n.skills = 3 et obs.size = 15

Tableau C.16 Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.16

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	200	15	0.316
Algorithme 2	3	24	200	15	0.316
Algorithme 3	3	24	200	15	0.315
Algorithme 4	3	24	200	15	0.325
Algorithme 5	3	24	200	15	0.288
Algorithme 6	3	24	200	15	0.314
Modèle IRT	3	24	200	15	0.289
Modèle valeurs attendues	3	24	200	15	0.313

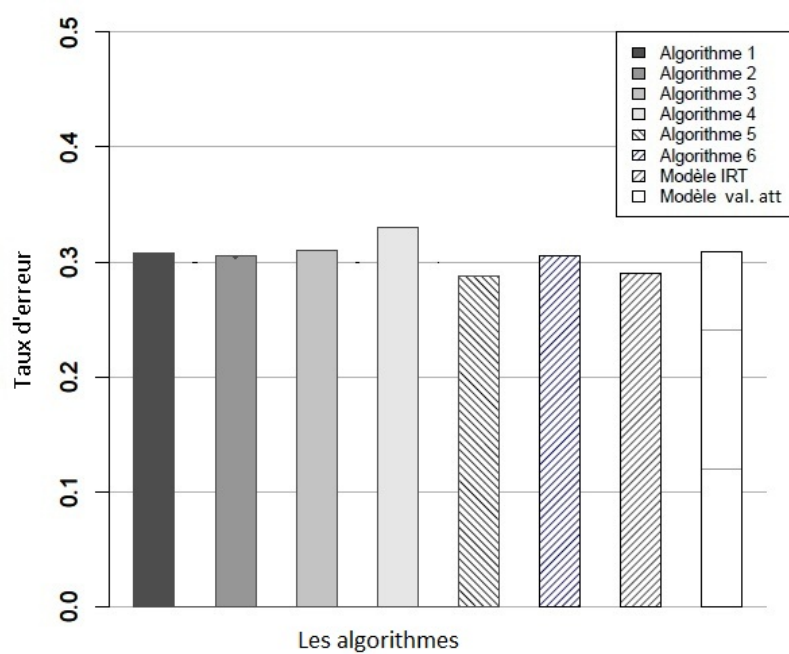


Figure C.17 Résultats de performance avec les données de Polytechnique avec le modèle conjonctif, n.skills = 3 et obs.size = 30

Tableau C.17 Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.17

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	200	30	0.307
Algorithme 2	3	24	200	30	0.305
Algorithme 3	3	24	200	30	0.310
Algorithme 4	3	24	200	30	0.330
Algorithme 5	3	24	200	30	0.287
Algorithme 6	3	24	200	30	0.305
Modèle IRT	3	24	200	30	0.290
Modèle valeurs attendues	3	24	200	30	0.308

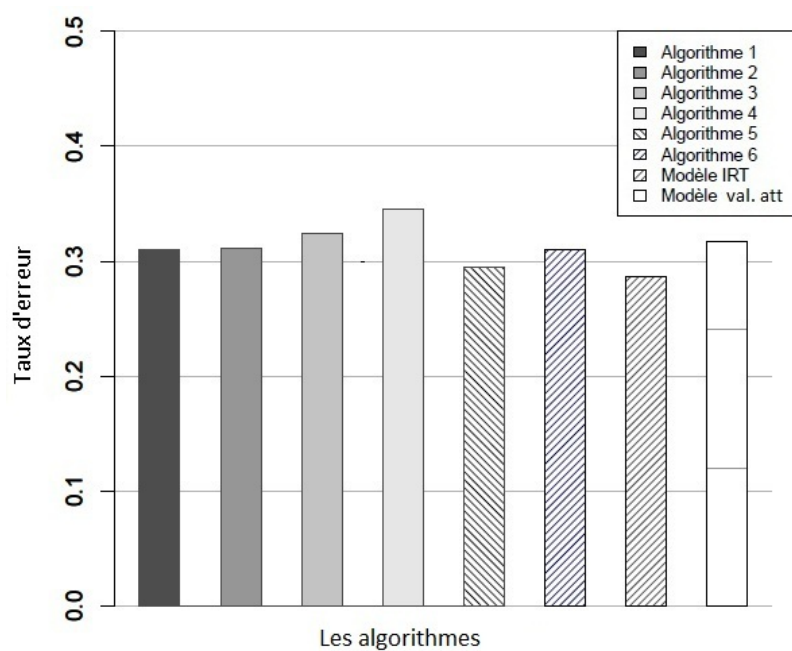


Figure C.18 Résultats de performance avec les données de Polytechnique avec le modèle conjonctif, n.skills = 3 et obs.size = 45

Tableau C.18 Tableau de valeurs de prédiction avec les données de Polytechnique, relatif à la figure C.18

Modèles	n.skills	n.run	train.size	obs.size	Taux d'err. moyen
Algorithme 1	3	24	200	45	0.310
Algorithme 2	3	24	200	45	0.311
Algorithme 3	3	24	200	45	0.324
Algorithme 4	3	24	200	45	0.345
Algorithme 5	3	24	200	45	0.294
Algorithme 6	3	24	200	45	0.310
Modèle IRT	3	24	200	45	0.286
Modèle valeurs attendues	3	24	200	45	0.316

ANNEXE D

Code R des algorithmes

Nous présentons dans cet annexe le code de nos algorithmes. La matrice **W** correspond à la matrice **Q** et la matrice **S** à la matrice des compétences. Finalement, la matrice **R** contient les résultats de tests.

Algorithme 1 :

```
Rhat.pred1 ← function(r, n1, n2, m1, m2, k){
  R.nmf ← nnmf(fix.sing(r[m1, n1]), k)
  W.m1k ← R.nmf$W
  H.kn1 ← R.nmf$H
  Wh.m2k ← r[m2, n1] % * %t(H.kn1) % * %solve(fix.sing((H.kn1) % * %t(H.kn1)))
  Hh.kn2 ← solve(t(W.m1k) % * %W.m1k) % * %t(W.m1k) % * %r[m1, n2]
  Rh.m2n2 ← Wh.m2k % * %Hh.kn2
  return(Rh.m2n2)
}
```

Algorithme 2 :

```
Rhat.pred2 ← function(r, n1, n2, m1, m2, k){
  Rm1.nmf ← nnmf(fix.sing(r[m1, n1]), k)
  Rm2.nmf ← nnmf(fix.sing(r[m2, n1]), k)
  W.m1k ← Rm1.nmf$W
  W.m2k ← Rm2.nmf$W
  H1.kn1 ← Rm1.nmf$H
  H2.kn1 ← Rm2.nmf$H
  Ph ← H2.kn1 % * %t(H1.kn1) % * %solve(fix.sing(H1.kn1 % * %t(H1.kn1)))
  Hh1.kn2 ← solve(fix.sing(t(W.m1k) % * %W.m1k)) % * %t(W.m1k) % * %r[m1, n2]
  Hh2.kn2 ← Ph % * %Hh1.kn2
  Rh.m2n2 ← W.m2k % * %Hh2.kn2
  return(Rh.m2n2)
}
```

Algorithme 3 :

```
Rhat.pred3 ← function(r, n1, n2, m1, m2, k){
  R.m2n1.nmf ← nnmf(fix.sing(r[m2, n1]), k)
  W.m2k ← R.m2n1.nmf$W
```

```

H.kn1 ← R.m2n1.nmf$H
Ph.km1 ← (H.kn1)% * %t(r[m1, n1])
Hh.kn2 ← Ph.km1% * %(r[m1, n2])
Rh.m2n2 ← (W.m2k% * %(Hh.kn2))
return(Rh.m2n2)
}

```

Algorithme 4 :

```

Rhat.pred4 ← function(r, n1, n2, m1, m2, k)
  Ph.m2m1
  ← (r[m2, n1])% * %t(r[m1, n1])% * %solve(fix.sing((r[m1, n1]))% * %t(r[m1, n1]))
  Rh.m2n2 ← Ph.m2m1% * %(r[m1, n2])
  return(Rh.m2n2)
}

```

Modèle IRT :

```

ltm.infer ← function(r, n1, n2, m1, m2, ...){
  r0 ← r[, n1]
  r.ltm ← ltm(t(r0) z1, control = list(method = "Nelder - Mead"))
  r2 ← t(r[, n2])
  r2[, m1] ← NA
  r2.fs ← factor.scores(r.ltm, r2)
  r.ltm.m2n2 ← apply(r2.fs$coef[m2, ], 1,
    function(i) plogis(r2.fs$score.dat$z1, i[1], 1/min(i[2]+1, 4)) )
  return(t(r.ltm.m2n2))
}

```

Algorithme 5 :

```

Rhat.pred.IRT ← function(r, n1, n2, m1, m2, k){
  m2n2 ← (ltm.infer(r, n1, n2, m1, m2))
  r2 j- r
  r2[rownames(m2n2), n2] j- m2n2
  R.nmf ← nnmf(fix.sing(r2), k)
  Rh ← R.nmf$W% * %R.nmf$H
  return(Rh[m2, n2])
}

```

Modèle des valeurs attendues :

```

PToOdds ← function(p)p/(1 - p)
OddsToP ← function(o)o/(1 + o)
Rhat.expect ← function(r, n1, n2, m1, m2, ...)
OddsToP(PToOdds(rowMeans(fix.sing(r[m2, n1]))))

```



```
%%o%%PToOdds(colMeans(fix.sing(r[m1,n2]))))
```

Algorithme 6 :

```
Rhat.pred.expt  $\leftarrow$  function( $r, n1, n2, m1, m2, k$ ){
  m2n2  $\leftarrow$  (Rhat.expect( $r, n1, n2, m1, m2$ ))
  r2  $\leftarrow$  r
  r2[rownames(m2n2),n2]  $\leftarrow$  m2n2
  R.nmf  $\leftarrow$  nnmf(fix.sing(r2), k)
  Rh  $\leftarrow$  R.nmf$W %*% R.nmf$H
  return(Rh[m2,n2])
}
```